



Universitat
Autònoma
de Barcelona



Projecte 5151 : PEÑA DE JUGADORES DE PRIMITIVA Y EUROMILLÓN

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica

realitzat per

David Carazo Herrera

i dirigit per

Ramón Grau Sala

Bellaterra, 18 de Juny de 2013



Universitat
Autònoma
de Barcelona



El sotasignat, **Ramón Grau Sala**

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en **David Carazo Herrera**

I per tal que consti firma la present.

Signat: **Ramón Grau Sala**

Bellaterra, 18 de Juny de 2013

RESUM

Mitjançant tecnologia JEE, farem una aplicació web segura (https) que ens permetrà gestionar una penya de persones que juga als sortejos de la Primitiva i l'Euromilió. Per una banda, les persones que s'hagin donat d'alta podran, mitjançant pagament previ, escollir amb quin o quins grups de la penya volen participar i gestionar la seva part privada. Per l'altra banda, l'administrador de la web disposarà d'un conjunt de menús que li permetran realitzar la gestió i el manteniment dels penyistes i de les participacions de la penya en els diferents sortejos en què es participi.

RESUMEN

Mediante tecnología JEE, haremos una aplicación web segura (https) que nos permitirá gestionar una peña de personas que juega en los sorteos de la Primitiva y el Euromillón. Por un lado, las personas que se hayan dado de alta podrán, mediante pago previo, escoger con que grupo o grupos de la peña quieren participar, y gestionar su zona privada. Por otro lado, el administrador de la web dispondrá de un conjunto de menús que le permitirán realizar la gestión y el mantenimiento de los peñistas y de las participaciones de la peña en los diferentes sorteos en que se participe.

SUMMARY

Through JEE technology, we'll build a secure web application that allow us to manage a group of people who play Primitiva and Euromillions lottery. On the one hand, registered persons may prior payment choose which lottery group want to participate, and manage the private area. On the other hand, the webmaster will have a set of menus for the management and maintenance of the group members and the lottery tips.

A la meva mare

Agraïments

Voldria donar les gràcies a la meva parella Susana, per llegir-se la memòria a mesura que anava ampliant-se i donar-me el seu punt de vista i els seus suggeriments, i sobretot pel seu suport incondicional, les seves mostres d'ànim, la seva confiança en mi i la seva comprensió al llarg de tot el projecte.

També vull donar les gràcies al meu tutor, Ramon Grau, pels bons consells que m'han ajudat a enfocar la solució d'alguns punts importants d'aquest projecte, i per les seves incansables paraules d'ànim.

Taula de continguts

1. Introducció	1
1.1 Motivacions.....	1
1.2 Objectius	1
1.3 Organització de la memòria	2
2. Fonaments teòrics	3
2.1 HTML	3
2.2 XHTML.....	3
2.3 JavaScript	4
2.4 Plataforma Java	4
2.5 JEE	5
2.6 Estat de l'art	6
3. Especificació de requeriments	8
3.1 Anàlisi de riscos	8
3.2 Recursos	9
3.3 Estimació de costos	10
3.4 Planificació temporal	11
4. Anàlisi	13
4.1 Diagrama de classes	13
4.1.1 Classes d'entitat (beans)	13
4.1.2 Classes gestores	16
4.1.3 Classes auxiliars.....	16

4.2	Diagrames de casos d'ús	18
4.2.1	Casos d'ús d'Usuari	18
4.2.2	Casos d'ús de Peñista.....	19
4.2.3	Casos d'ús d'Administrador	20
5.	Disseny	20
5.1	Capa client.....	22
5.2	Capa de presentació	22
5.3	Capa de negoci	23
5.4	Capa d'integració	24
5.5	Capa EIS (Enterprise Information System)	24
6.	Implementació	28
7.	Conclusions	38
8.	Annexos	41
A.	Glossari	41
B.	Descripció textual dels casos d'ús	54
C.	Arquitectura multicapa amb n-nivells (aplicació JEE).....	68
D.	Llibreries utilitzades a Eclipse	72
E.	Passos per fer l'aplicació web segura	73
	Bibliografia	75

Índex de figures

Figura 2.5.1	Esquema simplificat de contenidors i components de la plataforma JEE	6
Figura 4.1.1.1	Diagrama de classes d'entitat-beans	15
Figura 4.1.2.1	Diagrama de classes gestores	16
Figura 4.1.3.1	Diagrama de les classes auxiliars gestores del model de dades	17
Figura 4.1.3.2	Diagrama de la resta de classes auxiliars	17
Figura 4.2.1	Actors del sistema	18
Figura 4.2.1.1	Diagrama de casos d'ús d'Usuari	18
Figura 4.2.2.1	Diagrama de casos d'ús de Peñista	19
Figura 4.2.3.1	Diagrama de casos d'ús d'Administrador	20
Figura 5.1	Arquitectura de la nostra aplicació JEE no distribuïda	22
Figura 5.2.1	Diagrama de les JPS (vistes) de l'aplicació	23
Figura 5.5.1	Taula de penyistes	25
Figura 5.5.2	Taula d'apostes de la penya	25
Figura 5.5.3	Taula d'operacions realitzades a la penya	26
Figura 5.5.4	Taula de resultats dels sortejos	27
Figura 5.5.5	Taula dels grups de penyistes per cada tipus de sorteig	27
Figura 5.5.6	Taula del grups d'apostes amb les quals participa la penya als sorteigs	27
Figura 6.1	Diagrama reduït de consulta de tots els penyistes	29
Figura 6.2	Captura de pantalla d'un informe en format PDF i excel	30
Figura 6.3	Consulta de penyistes sense framework de paginació	31
Figura 6.4	Consulta de penyistes amb framework de paginació	31
Figura 6.5	Formulari amb validacions, calendari emergent i camp de captcha	32
Figura 6.6	Aplicació segura vista a Internet Explorer i Firefox	33
Figura 6.7	Pàgina de pagament	34
Figura 6.8	Pàgina de cobrament de premis i arxiu pdf generat	35
Figura 6.9	Menú de l'administrador sense framework de decoració	35
Figura 6.10	Menú de l'administrador amb framework de decoració	36
Figura g-4.1	Cicle de vida d'un servlet	43
Figura g-14.1	Relació els components principals de l'arquitectura JPA	51
Figura C.1	Arquitectura multicapa amb n-nivells	68
Figura C.2	Arquitectura JEE no distribuïda amb EJB's de sessió	71
Figura C.3	Arquitectura JEE distribuïda	71

1. Introducció

1.1 Motivacions

Un dels principals llasts del comerç electrònic és la falta de confiança del consumidor a Internet com a mitjà de compra i pagament. Aquest tipus de comerç presenta els mateixos problemes que altres sistemes de compra no presencial (com el telèfon o la compra per catàleg), en els quals les parts no contacten físicament. El comprador necessita tenir garanties sobre els productes/serveis que adquireix, el venedor garantia del pagament i hi ha d'haver un nivell de confidencialitat en la comunicació entre tots dos.

El nostre projecte ens permetrà aprendre nous coneixements en seguretat, i totes les tecnologies i eines necessàries per desenvolupar una aplicació web basada en tecnologia JEE.

1.2 Objectius

Les bases generals sobre les que han de cimentar-se les web que ofereixen serveis al sector de les loteries i apostes de l'estat són la seguretat, la confiança i la privacitat.

L'objectiu principal del nostre projecte és fer una d'aquestes webs de la mà d'una penya de jugadors que participa als sorteigs de la primitiva i l'euromilió, i tractarà de recrear aquests bases amb els mitjans disponibles.

Per garantir la seguretat farem servir protocols de servidors segurs ("https:...") amb certificats SSL de seguretat, d'aquesta manera la informació és encriptada abans de ser transferida via internet garantint la seva seguretat i privacitat. Crearem la nostra Autoritat Certificadora ^[g-1] (CA) que serà la que ens signarà el certificat del servidor.

Un parell de punts claus que tractarem seran:

- Cóm demostrar que un penyista que s'ha donat d'alta realment existeix?
- Quan paguem els premis, cóm se sap que aquests diners van a parar a una persona real que representa a aquest penyista?

La web serà accessible des de qualsevol navegador per Internet, i a més es podran veure un seguit d'estadístiques relatives a la penya per donar a l'usuari una visió el més completa possible.

1.3 Organització de la memòria

La memòria està dividida en una sèrie de capítols que corresponen, bàsicament, a les diferents etapes que formen el procés de desenvolupament d'aquest projecte.

Aquestes etapes són:

- Fonaments teòrics: Breu descripció de algunes de les tecnologies emprades i explicació de l'estat de l'art del projecte.
- Especificació de requeriments: En aquest apartat s'estudien els riscos, les característiques tècniques, el cost econòmic i la planificació temporal.
- Anàlisi: S'ha realitzat l'anàlisi del problema i el modelatge conceptual de la nostra solució mitjançant una sèrie de diagrames de classes i de casos d'ús.
- Disseny: S'han dissenyat tots els nivells de què consta la nostra aplicació web basada en tecnologia JEE (arquitectura, capes i taules de la base de dades)
- Implementació: Es detallen els canvis rellevant que ha sofert l'aplicació al llarg de les seves fases.
- Conclusions obtingudes del projecte.

Després d'aquestes etapes mostrem un conjunt d'annexos per completar certs aspectes tractats amb anterioritat i acabem amb la bibliografia utilitzada durant la realització del projecte.

2. Fonaments Teòrics

2.1 HTML

L'HTML (Hyper Text Markup Language) és el llenguatge de programació que s'utilitza per crear pàgines web. Un document HTML és un fitxer de text, amb extensió .htm o .html, amb un format especial que poden interpretar els diferents navegadors d'Internet.

Aquest llenguatge permet la creació de documents multimèdia amb text, imatges, taules i formularis per a la introducció de dades. A més permet definir enllaços a altres documents, permetent així la construcció d'hipertextos.

Un document HTML està format per text, que defineix el contingut del document, i per una sèrie d'etiquetes, que són les instruccions del llenguatge HTML, que defineixen la forma en la que s'ha de presentar la informació a l'usuari que navega per la web.

2.2 XHTML

XHTML (eXtensible Hyper Text Markup Language) és el llenguatge de marcat pensat per substituir a HTML com a estàndard per a les pàgines web. El seu objectiu és aconseguir una web semàntica, on la informació, i la forma de presentar-la estiguin clarament separades. En aquest sentit, XHTML serviria únicament per a transmetre la informació que conté un document, deixant per a fulles d'estil i JavaScript el seu aspecte i disseny en diferents mitjans (ordinadors, mòbils, ...)

2.3 JavaScript

JavaScript és el llenguatge de software més utilitzat a Internet per afegir dinamisme i interactivitat a les pàgines web.

No s'ha de confondre JavaScript amb Java, ja que aquest últim és un llenguatge de programació de propòsit general com també ho poden ser C++ o Visual Basic.

Un programa JavaScript s'integra dintre d'una pàgina web (entre el codi HTML), o en arxius independents amb extensió .js referenciats dins la mateixa, i és el navegador l'encarregat d'interpretar-lo i executar-lo. Tots els navegadors actuals interpreten el codi JavaScript integrat dintre les pàgines web.

JavaScript combinat amb DOM ^[g-2] i les fulles d'estil CSS ^[g-3] formen els fonaments del DHTML (Dynamic HTML) que permet a les aplicacions basades en navegador tornar-se molt més interactives.

A la part del client JavaScript permet millorar l'aspecte visual de les aplicacions (creació de finestres emergents, menús interactius, imatges que responen a accions de ratolí,...) i descarrega de feina al servidor (validacions de formularis des del client)

2.4 Plataforma Java

Sun ha organitzat la plataforma Java en tres dominis o àrees diferents:

- **Micro Edition (JME)** Plataforma de desenvolupament per a dispositius mòbils (PDA, telèfons mòbils, etc), Amb capacitat per executar Java.
- **Standard Edition (JSE)** Per executar desenvolupaments fets en Java en qualsevol ordinador que tingui instal·lada una màquina virtual Java.
- **Enterprise Edition (JEE)** Per desenvolupar i desplegar aplicacions empresarials que requereixin un conjunt de característiques complexes com seguretat, transaccionalitat, robustesa, alta disponibilitat, etc.

2.5 JEE

És una plataforma de desenvolupament empresarial, proposada per Sun Microsystems l'any 1997, que defineix un estàndard per al desenvolupament d'aplicacions empresarials multicapa (pot estar corrent en diversos ordinadors)

Defineix un model d'aplicacions distribuït, on podem dividir les aplicacions en parts i cadascuna es pot executar en servidors diferents, i multicapa amb n-nivells.

L'arquitectura JEE estableix un mínim de 3 capes:

- la capa client (p.e. en una aplicació web el browser),
- la capa intermèdia, i
- la capa de sistemes d'informació (EIS-Enterprise Information System)

JEE basa les aplicacions empresarials en el concepte de *components* de software modulars i estandarditzats que s'executen dins de *contenidors*. Els contenidors són entorns estàndard d'execució que proporcionen un conjunt de *serveis* de baix nivell als components que formen l'aplicació. Els servidors d'aplicacions són programes que implementen els serveis que ofereixen els contenidors als components, i hi ha de molts fabricants diferents, uns comercials i altres de codi lliure.

Tipus de components:

- *Components client*. Programa executat en entorn client (p.e. applet)
- *Components web* (servlets ^[g-4] i JSP's ^[g-5])
- *Components de negoci* (EJB's ^[g-6])

Tipus de contenidors:

- *Contenidor d'aplicacions* (el trobem a les màquines client amb aplicacions Java)
- *Contenidor d'applets* (el trobem en els navegadors de les màquines client)
- *Contenidor web* (el trobem al servidor)
- *Contenidor de EJB's* (el trobem al servidor)

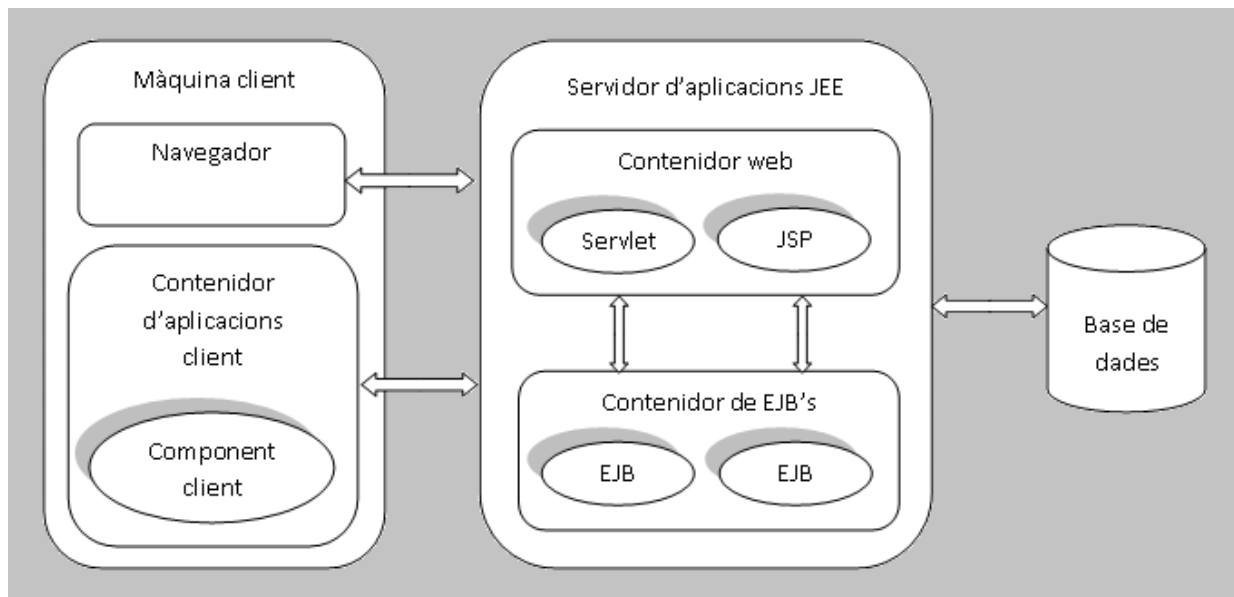


Figura 2.5.1 Esquema simplificat de contenidors i components de la plataforma JEE

Tipus de serveis:

- *Serveis configurables.* Permeten als desenvolupadors especificar el comportament dels components en temps de desplegat de l'aplicació (serveis de transaccions, serveis de seguretat, servei de noms)
- *Serveis no configurables* (gestió del cicle de vida dels components, gestió dels pools de connexions a recursos externs, accés a les API de JEE)

2.6 Estat de l'art

A l'hora de pagar per Internet amb targeta de crèdit, tant la validació com la realització efectiva del pagament es realitzen mitjançant el mateix sistema que en un comerç convencional. Una vegada que el número de targeta arriba al venedor, aquest l'envia fora d'Internet de la mateixa manera que en pagar en qualsevol botiga 'física'. Per tant, el punt crític es produeix quan s'envia el número de targeta a través d'una xarxa pública com és Internet.

Per garantir la confidencialitat en la transferència de dades a Internet s'utilitza el protocol de seguretat SSL ^[g-7] (Secure Sockets Layer), que accepten la immensa majoria de navegadors. Aquest sistema assegura la identitat del venedor, garanteix la

integritat del missatge i codifica les dades, el que garanteix que si algú intercepta la comunicació no podrà interpretar-la.

Per obtenir un certificat de seguretat s'ha de demanar a una Autoritat Certificadora, empreses dedicades a facilitar aquests tipus de certificats. El preu del certificat depèn de les característiques que posseeixi (tipus d'enciptació, garantia econòmica, EV SSL –barra de navegació verda-,...) i del temps que es desitja tenir-lo.

Però això no resol tots els problemes, ja que qualsevol persona amb accés a una targeta aliena pot pagar a Internet amb ella.

Per acabar amb aquestes limitacions s'han dissenyat nous estàndards de seguretat que a més certifiquen l'autenticitat de l'usuari de la targeta. Parlem del Comerç Electrònic Segur (CES), que afegeix un element més de seguretat mitjançant una clau associada a la targeta que només coneix el seu propietari. Així, després de teclejar el número de targeta en comprar en línia, el comerç sol licitarà al client, a través d'una comunicació amb l'emissor de la targeta, aquest codi personal que l'identifica com a legítim usuari de la targeta.

Hi ha bancs que en donar el servei d'alta demanen al client una clau secreta i una frase de seguretat, així en pagar en línia en un comerç segur amb targeta es mostra la frase de seguretat per garantir que la clau es tecleja en un lloc de confiança, i una vegada posada la clau si l'autenticació és correcta, s'autoritza la compra. Amb altres entitats al registrar-se al CES i haver donat d'alta el seu telèfon mòbil, cada vegada que es realitzi una compra per internet, es rebrà un SMS amb un codi de seguretat que es sol licitarà per completar la transacció.

Però el veritable cavall de batalla es troba en els petits pagaments (micropagaments), on les comissions fan que no compensi l'ús de targetes. Els sistemes per a micropagaments són necessaris en el mitjà virtual, on cada vegada hi ha més productes digitals amb preus reduïts (informes, diaris en PDF, música, vídeos, etc), o en el cas de les webs de jocs on el preu d'una aposta senzilla és petit.

Tot i l'existència de nombrosos sistemes de micropagaments, el més conegut és PayPal. Això és degut a la seva seguretat per realitzar transaccions en llocs web, a les moltes opcions de pagament que hi ha, a la possibilitat de fer-ho a través de dispositius mòbils, etc.

Algunes webs de jocs i apostes, a més de disposar d'aquests estàndards de seguretat, també estan auditades. L'auditoria permet assegurar a la web que en matèria jurídica-comptable està complint amb la legislació vigent, i contribueix a una correcta presa de decisions a través de la situació econòmic-financera de l'empresa.

També n'hi ha que són membres d'organitzacions que vetllen per la seguretat dels usuaris i les transaccions econòmiques per internet com: Adigital (www.adigital.org), Confianza Online (www.confianzaonline.es),...

Algunes de les webs consultades per veure el seu funcionament són:

- <https://www.lotopia.com>
- <https://www.ventura24.es>
- <https://www.hispaloto.es>
- <https://www.tuloto.com/>
- etc.

3. Especificació de requeriments

3.1 Anàlisi de riscos

Un cop feta una avaluació hem detectat els següents riscos que podrien afectar negativament al nostre projecte:

- Inadvertidament, esborrem o modifiquem erròniament algun fitxer clau.
- El nostre ordinador s'avaria.

Ja que no podem impedir que algun d'aquests riscos passi hem recorregut a les següents mesures per minimitzar l'impacte que podria causar la seva aparició durant el desenvolupament del nostre projecte:

- Fer còpies de seguretat setmanals del projecte i la documentació relacionada en un pendrive.
- Tenir muntat en un altre ordinador totes les aplicacions necessàries amb els fitxers i documents del projecte actualitzats per poder continuar desenvolupant la feina empresa.

3.2 Recursos

Aquest apartat pretén llistar els recursos de què es disposa, tant a nivell de hardware com a nivell de software, per a la realització del projecte.

Per tal de dur a terme el projecte hem fet servir un ordinador portàtil amb les següents prestacions:

- Processador Intel® Core™ i5.
- 4GB de RAM.
- 1.2GB d'espai lliure al disc dur.

També hem necessitat d'una connexió a internet (ADSL 12Mb) per poder fer proves, diverses consultes i trobar informació rellevant relativa al projecte.

Per a la realització d'aquest projecte han calgut com a recursos software:

- **Windows 7 Home Premium** Sistema operatiu de la màquina amb la qual hem fet aquest projecte. Llicència OEM per equips nous.
- **JDK 1.6.0_22** El kit de desenvolupament de Java és 'un conjunt de programes i llibreries distribuïdes gratuïtament que permeten desenvolupar, compilar i executar programes en Java.
- **Eclipse ^[g-8] IDE for JEE developers (versió Helios)** Entorn integrat de desenvolupament Java per crear aplicacions corporatives i aplicacions web. Software de codi obert amb llicència EPL (Eclipse Public License)

- **Tomcat** ^[g-9] **6.0** Servidor web amb suport de servlets i JSP's dins del qual s'allotgen tots els fitxers de la plataforma web. És un software open source.
- **MySQL** ^[g-10] **5.1.33** Sistema Gestor de Base de Dades necessari per crear i gestionar les taules del nostre projecte. Versió lliure amb llicència GPL.
- **MySQL GUI Tools 5.0** Les eines gràfiques tant per a l'administració del servidor MySQL com per interactuar amb les bases de dades.
- **iReport 3.7.1** Necessari per dissenyar els informes en format PDF, excel i word de la nostra aplicació. Es tracta de software lliure amb llicència AGPL (Affero General Public License).
- **OpenSSL 1.0** Eina de codi obert per implementar el protocol SSL, la fem servir per crear una CA i generar el certificat del servidor signat.
- **Microsoft® Office 2007** (aplicació Word) Per a la redacció de la memòria del projecte. Llicència de pagament, però si es disposa d'una adreça de correu a gmail es poden fer servir les aplicacions del google docs igualment.

3.3 Estimació de costos

Aquest no és un projecte d'empresa sinó que forma part dels crèdits obligatoris a realitzar per l'alumne, per tant el temps del projectista no suposa cap cost econòmic.

L'alumne ja té un equip informàtic que compleix els requeriments i disposa d'una connexió pròpia a Internet.

Tot el software utilitzat per a la realització del projecte, a excepció de l'emprat per realitzar la memòria, és de llicència gratuïta.

Tant el servidor web com la BD estan allotjats al portàtil abans citat per fer les proves de funcionament adients, però en el supòsit que s'hagués d'enregistrar un domini, i contractar l'allotjament web de l'aplicació java i la base de dades en MySQL, el cost anual rondaria els 90€. ^[1]

I respecte al certificat de confiança del nostre lloc web signat per la Autoritat Certificadora creada per nosaltres, si volguéssim comprar un certificat SSL per la nostra aplicació el cost anual seria d'uns 200€. [2]

3.4 Planificació temporal

A l'informe previ les fases previstes al projecte eren les següents:

- Fase I (12/11/2012 - 17/01/2013) Recerca d'informació addicional necessària per a la creació del projecte i per a la generació de l'informe previ.
- Fase II (18/01/2013 - 08/02/2013) Disseny lògica de negoci de les operacions CRUD (Create, Read, Update i Delete) dels participants de la penya, generació de taules a la BD, i realització de proves.
- Fase III (11/02/2013 - 27/02/2013) Disseny lògica de negoci dels jocs de la primitiva i l'euromilió, generació i actualització de taules a la BD, i realització de proves
- Fase IV (28/02/2013 - 30/04/2013) Conversió a projecte web dinàmic (servlets, JSP, javascript,...), generació d'informes amb tecnologia JasperReport, inserció de framework de persistència basat en JPA, seguretat i realització de proves.
- Fase V (01/05/2013 - 21/05/2013) En cas que no haguem acabat alguna part en els terminis programats encara disposem de temps per finalitzar el projecte abans del període de sol·licitud de lectura. En cas de temps extra afegirem més funcionalitats.

Segons aquesta planificació la data de finalització era el 21 de maig de 2013, i es destinaria el temps restant fins a la data de lliurament de la memòria per a la realització de la mateixa. Es van estimar unes 499 hores de feina (exceptuant vacances i festius), suposant 3 hores de treball diari a la fase I, 4 hores diàries a les fases II,III i IV, i 8 hores diàries a la fase V.

La planificació inicial ha variat substancialment degut a canvis en els objectius del projecte i complicacions en l'apartat de seguretat que finalment han estat resolt:

- Durant la fase IV es va decidir afegir la possibilitat de veure estadístiques que reflectissin dades de la penya, la qual cosa va provocar haver de redissenyar la nostra BD i implementar el codi per afegir gràfiques a la nostra aplicació mitjançant el framework JfreeChart. També es va afegir paginació per poder veure les consultes a la BD d'una manera més còmoda.
- En el transcurs de la fase IV vam tenir alguns problemes amb la generació dels nostres propis certificats de confiança que vam arrossegar fins al final de la fase V.
- A la fase V també es va incorporar un framework de decoració a la web per donar-li una aparença més professional, i es va afegir la modalitat de pagament per part dels penyistes via PayPal. Motius pels quals es va allargar aquesta fase fins el 29 de Maig de 2013.

De manera que les hores de feina resultants es reflecteixen a la següent taula:

	Previsió inicial hores	Hores reals	
Fase I (29 dies)	87	87	
Fase II (16 dies)	64	66	(+ 2)
Fase III (13 dies)	52	53	(+ 1)
Fase IV (44 dies)	176	248	(+ 72)
Fase V (15 dies)	120	161	(+ 41)
TOTAL	499	615	

Taula 3.5.1: Temps estimat i real de les fases del projecte

Com es pot veure hi ha hagut un increment del 23% en el temps de treball respecte al inicialment planificat degut als motius abans esmentats. I encara que hem hagut de dedicar més hores hem pogut acabar les fases dins de les dates establertes per poder presentar-nos a la convocatòria de juny.

4. Anàlisi

Per al desenvolupament de software orientat a objectes (OO) no hi ha prou amb fer servir un llenguatge OO. També és necessari realitzar un anàlisi i disseny OO. Per aquest motiu i perquè la nostra aplicació està feta en un llenguatge OO, com és el Java, utilitzarem el model conegut com *Unified Modeling Language* (UML).

L'UML és el model per a la construcció de software OO que l'organització OMG ^[3] ha proposat com un estàndard d'ISO, i que des dels últims anys s'ha imposat a tot el món. Consta d'un conjunt de diagrames interrelacionats que serveixen per descriure els diversos aspectes de l'estructura i la dinàmica del software, i ens permeten comprendre millor l'aplicació, el que ens facilitarà la tasca d'implementació, així com l'ampliació del sistema si fos necessari.

En el nostre cas ens centrarem en dos tipus de diagrames:

- Diagrama de classes
- Diagrama de casos d'ús

4.1 Diagrama de classes

4.1.1 Classes d'entitat (beans)

Tenim les següents classes d'entitat-beans, detallades amb els seus atributs i mètodes:

- *Peñista*: Representa a un penyista que forma part de la penya. És l'encarregada d'encapsular un objecte de tipus 'Peñista' del model de dades.
- *Operacion*: Representa una participació d'un dels grups de la penya a un sorteig. S'ha previst, en futures millores, la possibilitat de permetre als penyistes fer apostes individuals, de manera que la classe està preparada per si es volgués fer aquest pas. És l'encarregada d'encapsular un objecte de tipus 'Operación' del model de dades.

- *Apuesta*: Representa a una aposta (primitiva o euromilió) que pot formar part o no del grup d'apostes que s'hi juguen als sorteigs. És l'encarregada d'encapsular un objecte de tipus 'Apuesta' del model de dades.
- *Sorteo*: Representa un sorteig que ha passat i l'administrador ha emmagatzemat, encara que la penya no hagi participat pel motiu que sigui. Formarà part de l'històric de sorteigs a la penya. Servirà per contrastar el resultat amb el grup d'apostes que coincideixin amb la data de sorteig. Encarregada d'encapsular un objecte de tipus 'Sorteo' del model de dades.
- *HistorPen*: Representa el bloc de penyistes que formen part dels grups de la penya en el mes actual. Si es treu o s'afegeix un penyista d'algun grup el número de referència del bloc s'actualitza incrementant-se. És l'encarregada d'encapsular un objecte de tipus 'HistorPen' del model de dades.
- *HistorApu*: Representa el bloc d'apostes que fa servir la penya per participar en els sortejos (primitiva o euromilió) del mes actual. Si es treu o s'afegeix una aposta d'algun grup el número de referència del bloc s'incrementa. És l'encarregada d'encapsular un objecte de tipus 'HistorApu' del model de dades.

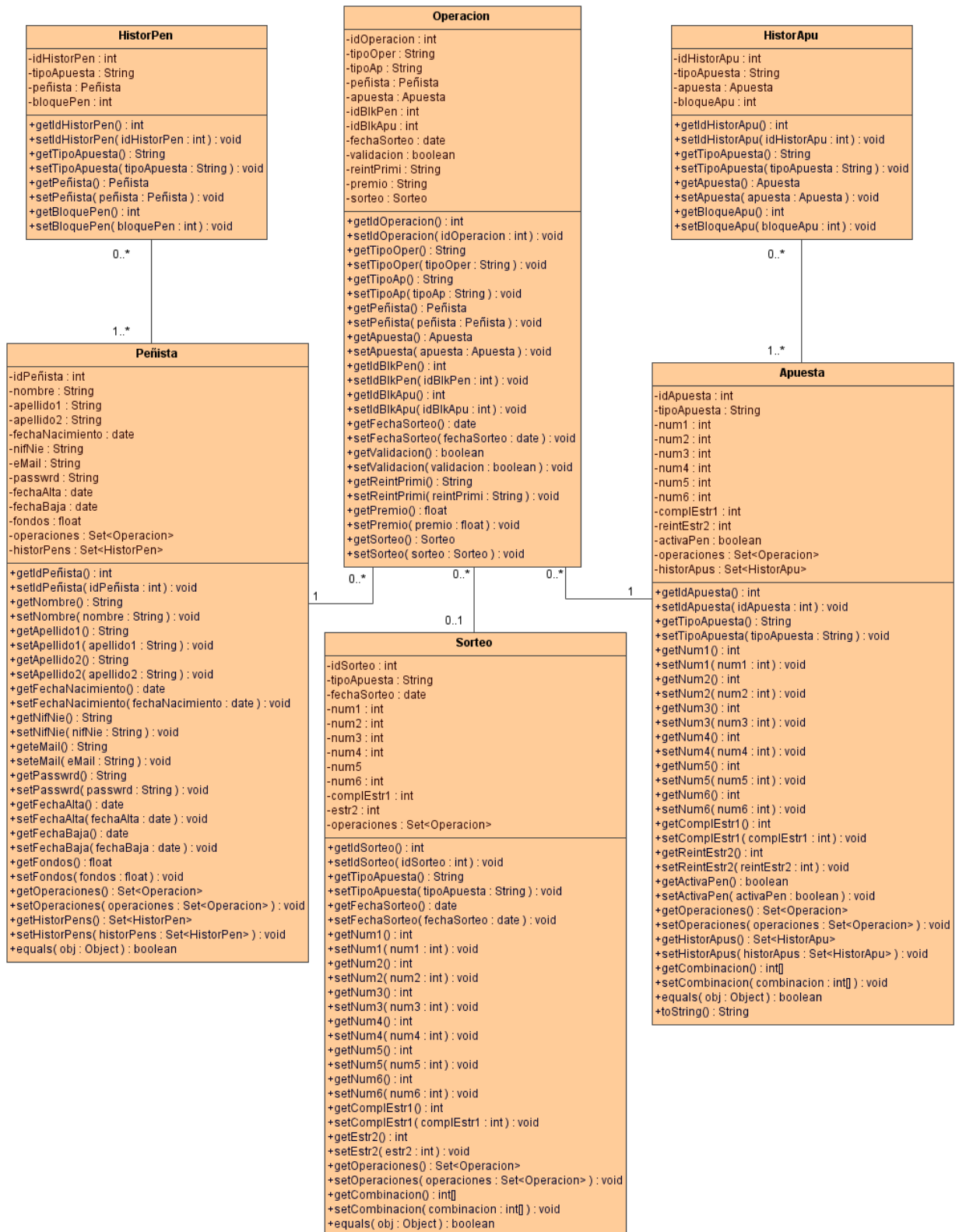


Figura 4.1.1.1 Diagrama de clases d'entitat-beans

4.1.2 Classes gestores

Aquestes classes s'han implementat com servlets, i ens serveixen per gestionar l'aplicació:

- *Controlador*: Gestiona les peticions entre els clients i les pàgines JSP, executa la lògica de negoci adequada en base a la petició que li arriba i s'encarrega de la selecció de la següent vista.
- *Report*: Encarregat de generar i presentar els informes amb la tecnologia JasperReport.

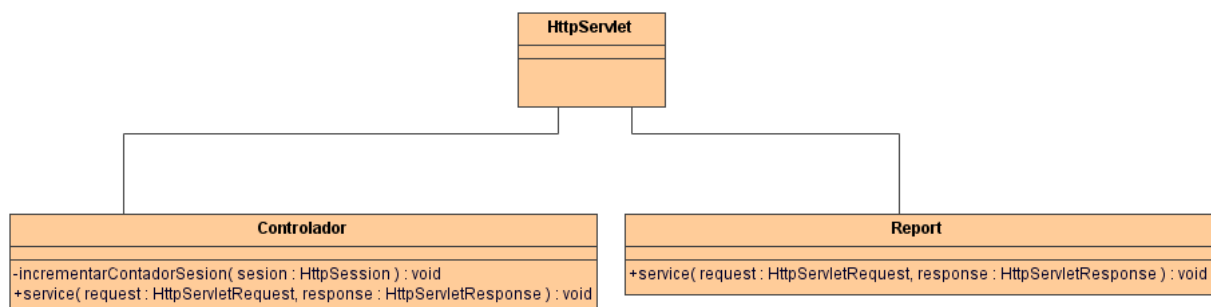


Figura 4.1.2.1 Diagrama de classes gestores

4.1.3 Classes auxiliars

Les següents classes ens permeten gestionar la persistència de les classes d'entitat abans esmentades:

- *GestPeñista*: Persisteix, materialitza, i recupera objectes de tipus Peñista.
- *GestApuesta*: Persisteix, materialitza, i recupera objectes de tipus Apuesta.
- *GestOperacion*: Persisteix, materialitza, i recupera objectes de tipus Operacion.
- *GestSorteo*: Persisteix, materialitza, i recupera objectes de tipus Sorteo.
- *GestHistorPen*: Persisteix, materialitza, i recupera objectes de tipus HistorPen.
- *GestHistorApu*: Persisteix, materialitza, i recupera objectes de tipus HistorApu.

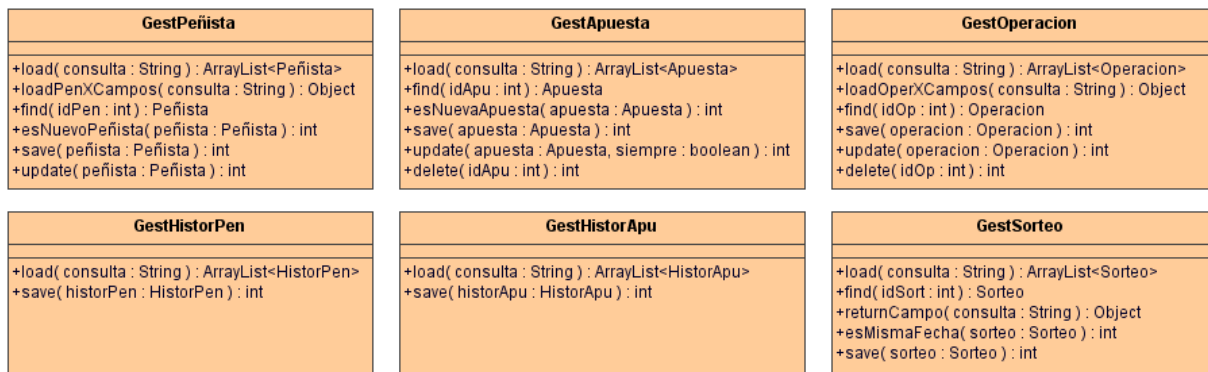


Figura 4.1.3.1 Diagrama de les classes auxiliars gestores del model de dades

I amb les següents classes proveïm a les classes gestores dels mètodes necessaris per la lògica de negoci de l'aplicació, i generem les gràfiques que surten a la secció d'estadístiques:

- *Metodo*: Porta els mètodes més importants que cridarà el servlet controlador per generar la lògica de negoci de l'aplicació.
- *Graficas*: Disposa dels mètodes necessaris per generar les gràfiques que sortiran en la secció d'estadístiques.
- *Conexión*: Ens permet generar un object de tipus Connection per poder connectar-nos i desconnectar-nos de la BD.

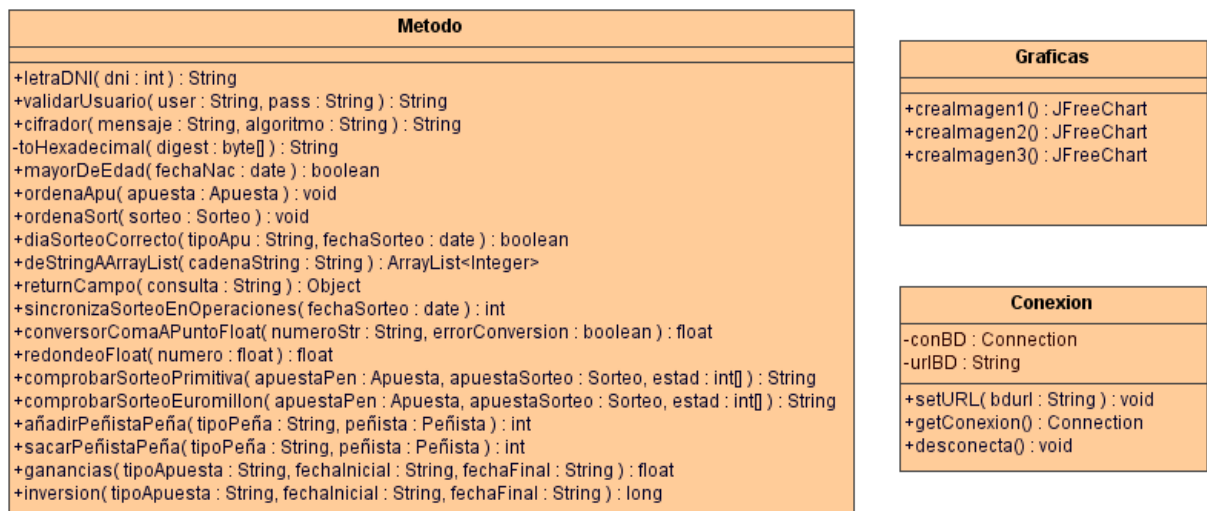


Figura 4.1.3.2 Diagrama de la resta de classes auxiliars

4.2 Diagrames de casos d'ús

Els diagrames de casos d'ús ens permetran conèixer els diferents tipus de actors que podran accedir i / o utilitzar la nostra aplicació. També ens servirà per especificar les accions que tindran aquests usuaris dins de la nostra aplicació, així com les relacions que apareixeran entre els diferents casos d'ús.

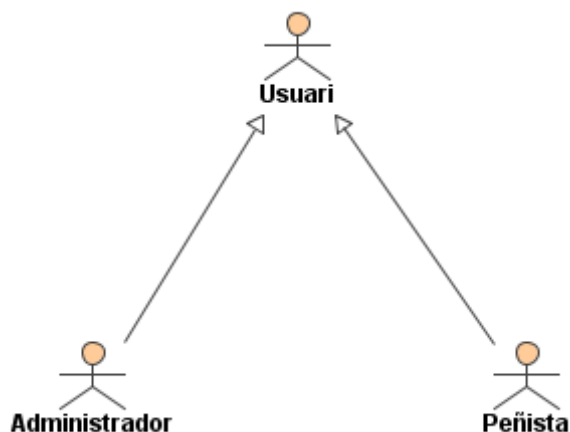


Figura 4.2.1 Actors del sistema

A la Figura 4.2.1 es pot veure els diferents tipus d'actors que podran accedir a la aplicació. Tots ells deriven d'Usuari. A continuació es detalla el diagrama dels casos d'ús de cadascun dels actors. Es pot veure la descripció textual de tots els casos a l'*annex-B* d'aquesta memòria.

4.2.1 Casos d'ús d'Usuari

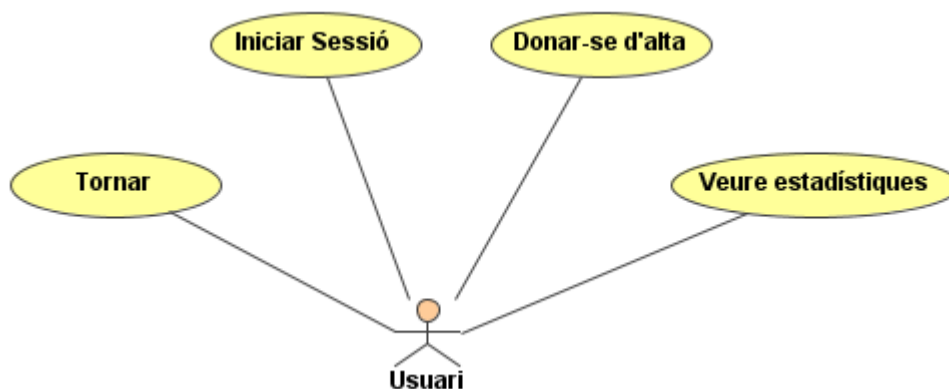


Figura 4.2.1.1 Diagrama de casos d'ús d'Usuari

Aquest tipus d'actor, que també podria dir-se usuari anònim o visitant, no s'ha identificat davant del sistema, ja sigui perquè encara no ha iniciat la sessió o perquè està de pas o cercant informació. Les accions que podrà realitzar seran molt limitades, centrant-se en l'obtenció d'informació estadística de la penya. No obstant això, pot accedir a la zona d'enregistrament o a l'apartat de login per autenticar-se i entrar a la zona privada de l'aplicació, accedint, per tant, a una major funcionalitat, si és que disposa d'un nom d'usuari i contrasenya vàlids.

4.2.2 Casos d'ús de Peñista

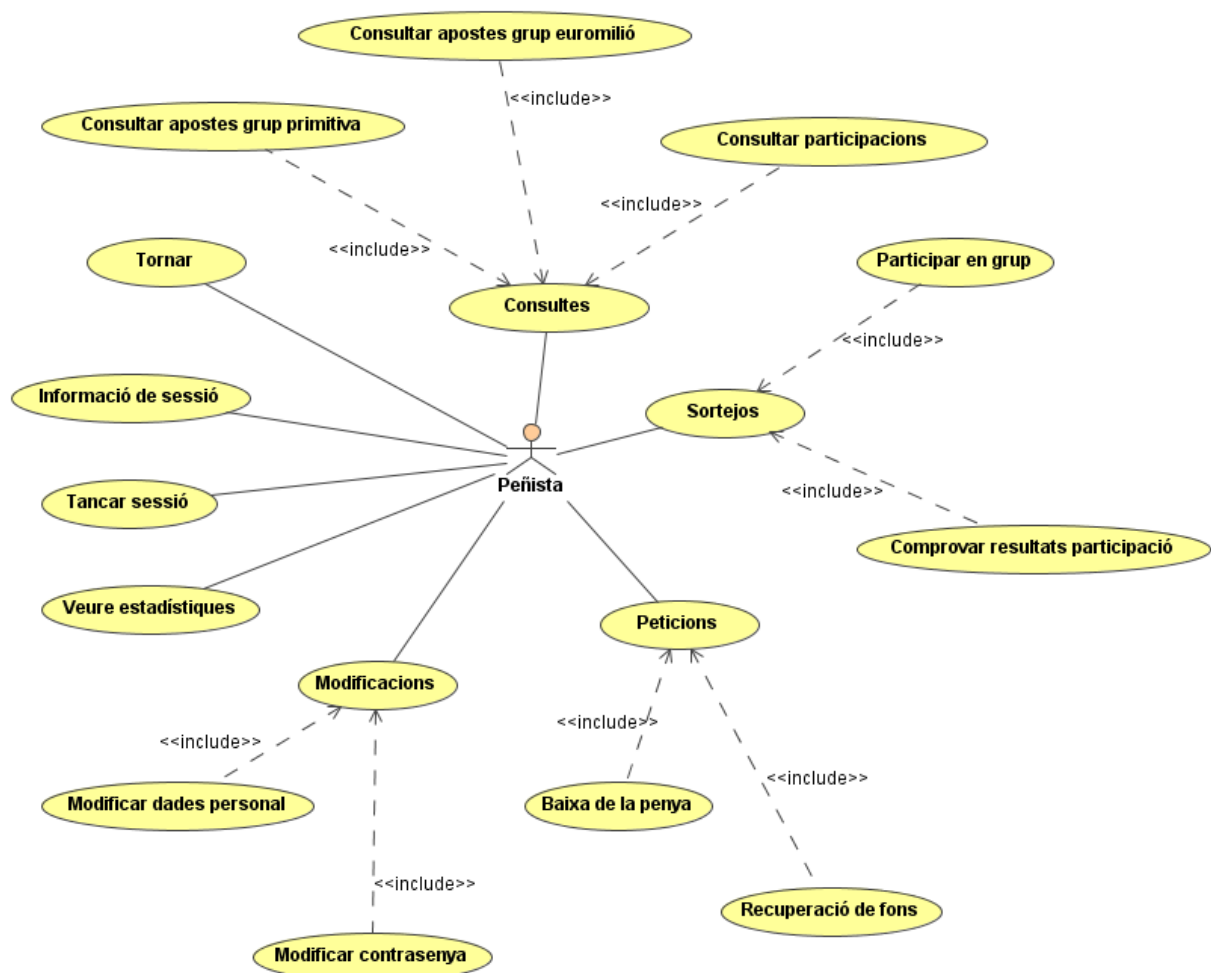


Figura 4.2.2.1 Diagrama de casos d'ús de Peñista

Aquest tipus d'actor representa un penyista enregistrat a la penya.

4.2.3 Casos d'ús d'Administrador

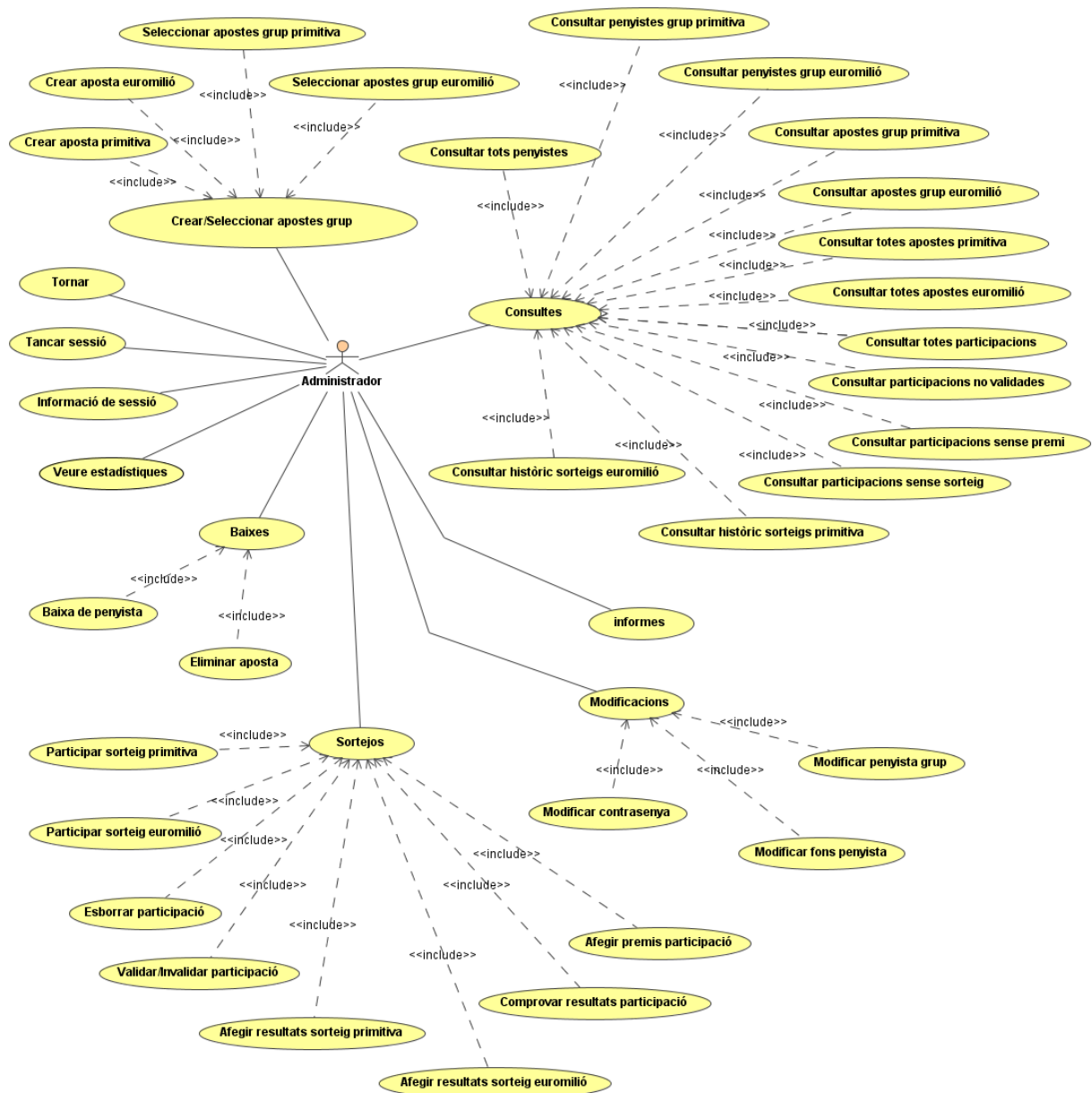


Figura 4.2.3.1 Diagrama de casos d'ús d'Administrador

Aquest tipus d'actor representa l'administrador encarregat de gestionar la penya.

5. Disseny

Des del punt de vista de l'Enginyeria del Software, la divisió d'un sistema en capes facilita el disseny modular (cada capa encapsula un aspecte concret del sistema) i permet la construcció de sistemes dèbilment acoblats (si minimitzem les

dependències entre capes, resultarà més fàcil substituir l'implementació d'una capa sense afectar a la resta del sistema)

Una aplicació JEE es pot dividir en 3 capes d'alt nivell, que normalment es subdivideixen en 5 capes lògiques (a l'*annex-C* es pot veure una descripció més detallada de l'arquitectura multicapa amb n-nivells d'una aplicació JEE):

- Capa client
- Capa intermèdia, dividida en 3 subcapes:
 - Capa de presentació
 - Capa de negoci
 - Capa d'integració
- Capa EIS

La nostra aplicació és una aplicació **no distribuïda**, perquè la capa intermèdia i tots els components que la formen es troben desplegats a la mateixa màquina, de fet, en el mateix servidor. Nosaltres farem servir una arquitectura amb components no distribuïts i sense utilitzar EJB's per la capa de negoci.

A grans trets direm que:

- Aquesta és l'arquitectura més simple, on els components web i els de negoci s'executen en la mateixa màquina virtual i els components de negoci no són EJB.
- Només es necessita un contenidor web amb capacitat de servlets (Tomcat)
- Malgrat la seva senzillesa, hi ha una clara separació entre els components de negoci i els components de presentació.

En aquesta arquitectura es té:

- Capa de presentació amb servlets i JSP's seguint un patró de disseny MVC.
- Accés local de la capa de presentació a la capa de negoci.
- Capa de negoci implementada mitjançant classes POJO ^[g-11].
- Capa d'integració desenvolupada amb un framework de persistència, o motor ORM ^[g-12].

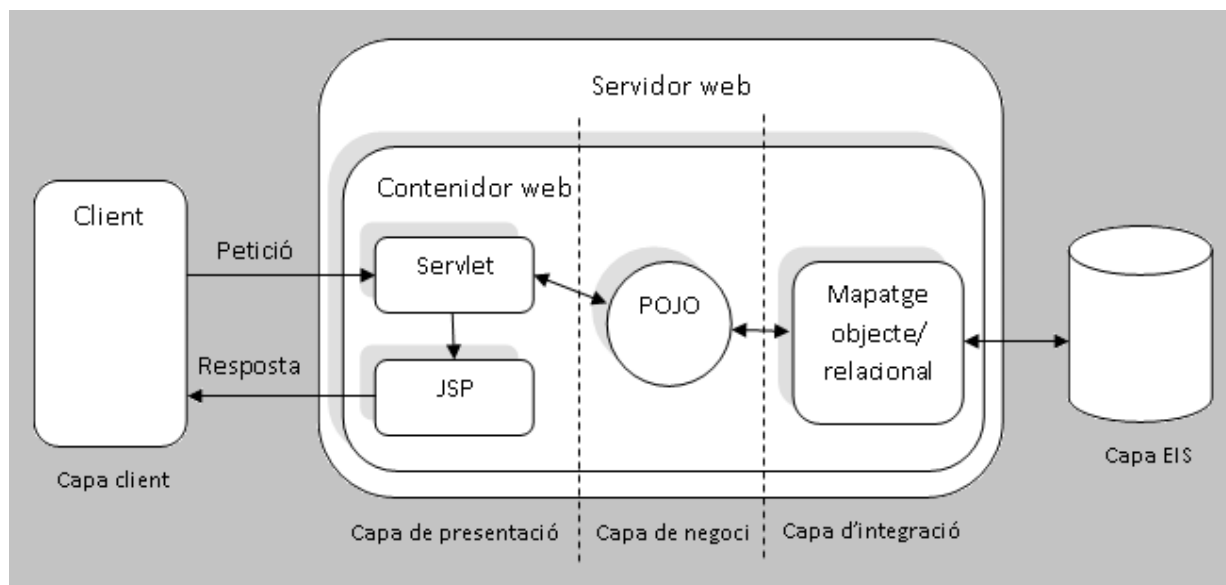


Figura 5.1 Arquitectura de la nostra aplicació JEE no distribuïda

5.1 Capa client

És molt important fer una bona elecció del tipus de client que es proporciona amb una aplicació, ja que és la "cara" de l'aplicació, la part amb la que interactuarà l'usuari. En el nostre cas tractarem amb els clients web (thin client)

5.2 Capa de presentació

Per a la nostra aplicació web farem servir el patró de disseny MVC (Model Vista Controlador), també anomenat Model-2. Les aplicacions basades en aquest model:

- Permeten una separació clara de rols a la capa de presentació (divisió de les responsabilitats)
- Són més flexibles i fàcils de mantenir,
- Permeten gestionar centralitzadament totes les peticions (per exemple aplicar seguretat i logs)

El servlet "controlador" s'encarregarà d'executar la *lògica de negoci* adequada en base a la petició que li arribi. També s'encarregarà de centralitzar la *lògica de selecció de la següent vista* en base a:

- Els paràmetres d'entrada,

- El resultat de la invocació de la lògica de negoci, i/o
- L'estat de l'aplicació (flux normal, error, etc)

Farem servir les JSP's per generar les vistes en format textual amb què interactuaran els usuaris (layout de presentació). I per intentar minimitzar la quantitat de codi java present a les JSP's (scriptlets) utilitzarem les llibreries estàndard d'etiquetes JSTL (Java Standard Tag Library) [g-13]

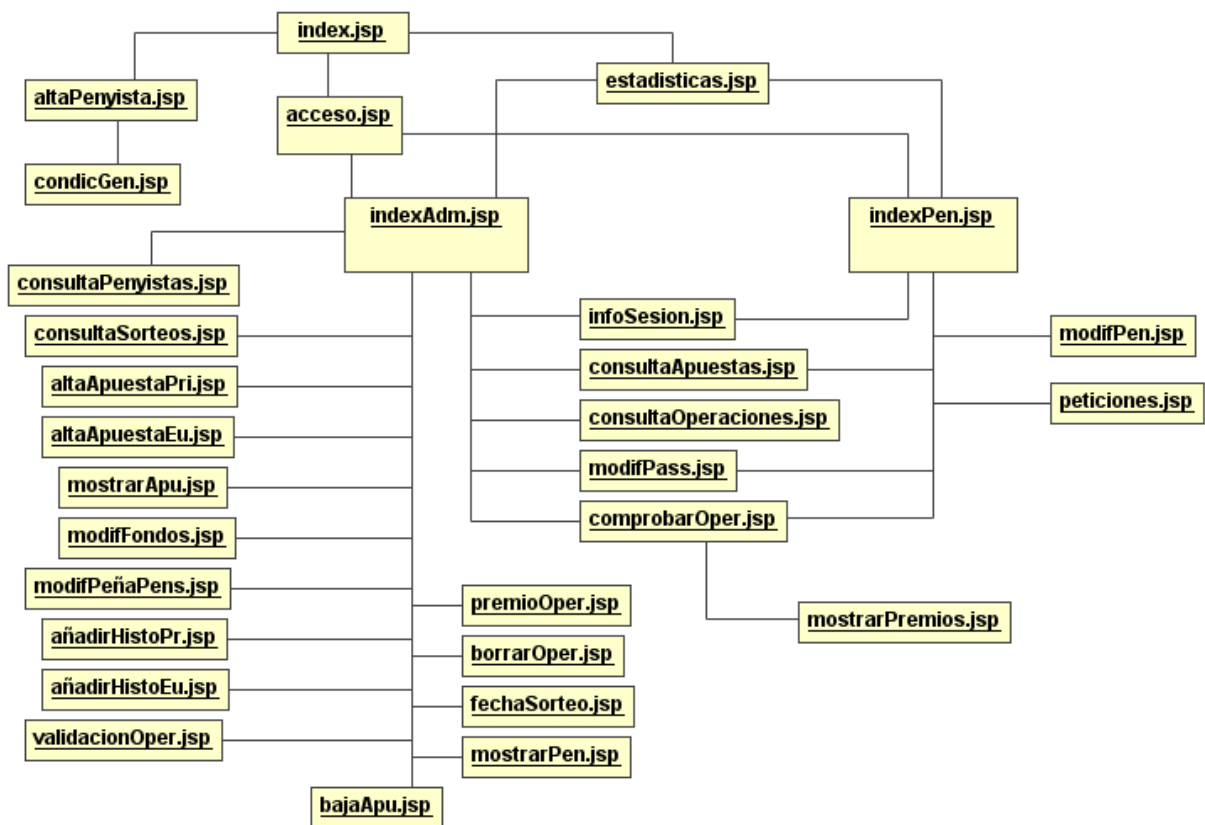


Figura 5.2.1 Diagrama de les JPS (vistes) de l'aplicació

5.3 Capa de negoci

Aquesta capa proporciona les funcionalitats de negoci específiques del nostre projecte web (xifrat de la contrasenya, comprovació del resultat d'un sorteig, etc)

En el cas que tinguéssim molts usuaris, els EJB's de sessió serien una bona opció com a components de negoci per a aquesta capa, però com hem dissenyat l'aplicació suposant que tindrem pocs usuaris utilitzarem classes POJO.

Les possibles solucions que hem trobat respecte a la quantitat d'usuaris que accedeixen a la nostra aplicació són:

- pocs usuaris → classes POJO
- molts usuaris → classes EJB, o classes POJO amb algun framework com per exemple Spring.

5.4 Capa d'integració

La divisió de la capa intermèdia entre capa de negoci i capa d'integració és purament lògica (artificial), ja que els components de les dues capes estan físicament a la mateixa màquina i desplegats dins del mateix contenidor.

Per automatitzar les operacions CRUD (creació, lectura, actualització i esborrat) de les entitats, o dades, en la nostra BD relacional, farem servir classes POJO conjuntament amb el framework de persistència, basat en JPA ^[g-14], anomenat EclipseLink.

El principal avantatge d'aquestes eines és la facilitat d'ús, la persistència transparent d'entitats, moltes optimitzacions de rendiment, generació automàtica de classes java que representen les entitats i les relacions a partir de fitxers de definició, etc.

5.5 Capa EIS (Enterprise Information System)

A continuació veurem les taules de la BD relacional *dbpenya*, ja que els noms dels camps són prou descriptius donarem una petita explicació dels camps més característics:

- A la taula **penyistas** la clau primària és l'identificador de penyista. Respecte a la resta d'atributs direm que el camp *nif_nie* és el login d'entrada a l'aplicació web, el password es guardarà encriptat a la BD amb l'algorisme SHA-1, i el camp *fondos* és el fons acumulats dels premis.

Table Name:	penyistas	Database:	dbpenya	Comment:	
Columns and Indices					
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id_penyista	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
nombre	VARCHAR(50)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
apellido_1	VARCHAR(50)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
apellido_2	VARCHAR(50)			<input type="checkbox"/> BINARY	NULL
fecha_nacimiento	DATE	<input checked="" type="checkbox"/>			NULL
nif_nie	VARCHAR(9)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
e_mail	VARCHAR(100)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
passwd	VARCHAR(50)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
fecha_alta	DATE	<input checked="" type="checkbox"/>			NULL
fecha_baja	DATE				NULL
fondos	FLOAT	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figura 5.5.1 Taula de penyistes

- Pel que fa a la taula **apuestas** la clau primària és l'identificador d'aposta. El camp *tipo_apuesta* pot ser 'primitiva' o 'euromilió' i el camp *activa_pen* reflexa si l'aposta forma part del grup d'apostes actuals que es fan servir per participar als sorteigs. Per representar una aposta primitiva, amb una combinació de 6 números, farem servir dels camps *num_1* a *num_6*, i a l'inici tindrem un '-1' com a valor de control als camps complementari (perquè no el sabrem fins que no es faci el sorteig) i reintegre (perquè no el sabrem fins que no es faci la validació a l'administració). Pel que fa a l'euromilió el camp *num_6* tindrà el valor de control '-1' perquè la combinació consta de 5 números.

Table Name:	apuestas	Database:	dbpenya	Comment:	
Columns and Indices					
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id_apuesta	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
tipo_apuesta	VARCHAR(10)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
num_1	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_2	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_3	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_4	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_5	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_6	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
compl_estr_1	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
reint_estr_2	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
activa_pen	TINYINT(1)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figura 5.5.2 Taula d'apostes de la penya

- La taula **operaciones** té de clau primària l'identificador d'operació (participació). El camp *tipo_operación* que pot ser 'individual' o 'multiple' està creat per la possible millora de permetre als penyistes jugar apostes senzilles. Els camps *id_penyista*, *id_apuesta* i *id_sorteo* són claus foranies de les taules penyistas, apuestas i sorteos. El camp *id_blk_pen* és el número que fa referència al bloc de penyistes que hi juguen en un sorteig actual, aquest camp té molt a veure amb el camp *bloque_pen* de la taula historpen. El camp *id_blk_apu* és el número que fa referència al bloc de apostes actual amb les quals es juga en un tipus de sorteig concret, aquest camp es troba molt associat amb el camp *bloque_apu* de la taula historapu. El camp *validació* l'activa l'administrador un cop s'hagi pagat l'aposta a l'administració, el camp *reint_primi* s'omplirà quan se sàpiga que reintegraments ens ha donat l'administració al validar-se i el camp *premio* es posarà quan, un cop ha sortit el resultat del sorteig, sapiguem a com es paga la categoria de premis.

Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value	Comment
id_operacion	INT(10)	✓	✓	UN SIGNED ZEROFILL	NULL	
tipo_operacion	VARCHAR(10)	✓		BINARY	NULL	
tipo_apuesta	VARCHAR(10)	✓		BINARY	NULL	
id_penyista	INT(10)			UN SIGNED ZEROFILL	NULL	
id_apuesta	INT(10)			UN SIGNED ZEROFILL	NULL	
id_blk_pen	INT(10)			UN SIGNED ZEROFILL	NULL	
id_blk_apu	INT(10)			UN SIGNED ZEROFILL	NULL	
fecha_sorteo	DATE	✓			NULL	
validacion	TINYINT(1)	✓		UN SIGNED ZEROFILL	NULL	
reint_primi	VARCHAR(25)	✓		BINARY	NULL	
premio	FLOAT	✓		UN SIGNED ZEROFILL	NULL	
id_sorteo	INT(10)			UN SIGNED ZEROFILL	NULL	

Figura 5.5.3 Taula d'operacions realitzades a la penya

- El camp identificador de sorteig és la clau primària de la taula **sorteos**. Per representar el resultat d'un sorteig de primitiva farem servir dels camps *num_1* a *num_6* per emmagatzemar la combinació, el camp *compl_estr_1* pel complementari i *reint_estr_2* pel reintegrament. I pel que fa a l'euromilió, el camp *num_6* tindrà el valor de control '-1', i emmagatzemem les estrelles als camps *compl_estr_1* i *reint_estr_2*.

Table Name:	sorteos	Database:	dbpenya	Comment:	
Columns and Indices					
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id_sorteo	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
tipo_apuesta	VARCHAR(10)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
fecha_sorteo	DATE	<input checked="" type="checkbox"/>			NULL
num_1	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_2	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_3	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_4	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_5	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
num_6	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
compl_estr_1	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
reint_estr_2	INT(11)	<input checked="" type="checkbox"/>		<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figura 5.5.4 Taula de resultats dels sortejos

- Per la taula **historpen** la clau primària es el camp *id_histor_pen*, el camp *id_penyista* és clau forania de la taula penyistas i el camp *bloque_pen* identifica cada penyista amb el bloc que ha jugat o hi juga actualment.

Table Name:	historpen	Database:	dbpenya	Comment:	
Columns and Indices					
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id_histor_pen	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
tipo_apuesta	VARCHAR(10)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
id_penyista	INT(10)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
bloque_pen	INT(10)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figura 5.5.5 Taula dels grups de penyistes per cada tipus de sorteig

- Per acabar la taula **historapu** té com a clau primària el camp *id_histor_apu*, el camp *id_apuesta* és clau forania de la taula apuestas i el camp *bloque_apu* identifica cada aposta amb el bloc d'apostes que s'han fet al llarg de la creació de la penya.

Table Name:	historapu	Database:	dbpenya	Comment:	
Columns and Indices					
Column Name	Datatype	NOT NULL	AUTO INC	Flags	Default Value
id_histor_apu	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
tipo_apuesta	VARCHAR(10)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY	NULL
id_apuesta	INT(10)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL
bloque_apu	INT(10)	<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL	NULL

Figura 5.5.6 Taula del grups d'apostes amb les quals participa la penya als sorteigs

6. Implementació

Durant la fase II, on encara no teníem un projecte web dinàmic, vam generar les classes necessàries per poder realitzar les operacions CRUD amb la taula penyistas de la BD, mitjançant un senzill menu que es visualitzava a través de la consola d'Eclipse. No hi havia interfície gràfica i encara no teníem cap framework de persistència, l'accés a la BD es feia mitjançant JDBC ^[g-15] directament.

Al final de la fase III ja teníem un prototip funcional bàsic de la nostra aplicació basat en classes java per poder testar la lògica de negoci de la penya i la gestió dels penyistes, però seguia sense ser una aplicació web i tractàvem la persistència amb JDBC pur.

A partir de la fase IV va canviar el projecte totalment a l'hora d'introduir-nos en la tecnologia JEE. Es va configurar Tomcat des del propi Eclipse i vam crear un projecte web dinàmic. A partir d'aquí per poder treballar amb els frameworks i tecnologies que necessitàvem només havíem d'incloure en el buildpath del nostre projecte les llibreries corresponents, arxius amb extensió '. jar' (es poden veure a l'*annex-D* totes les llibreries incorporades)

A continuació tractarem breument els canvis rellevant que es van fer a la nostra aplicacio web dinàmica:

Servlets i JSP's

La figura següent il·lustra el funcionament resumit de la consulta d'un usuari de tots els penyistes que formen part de la penya:

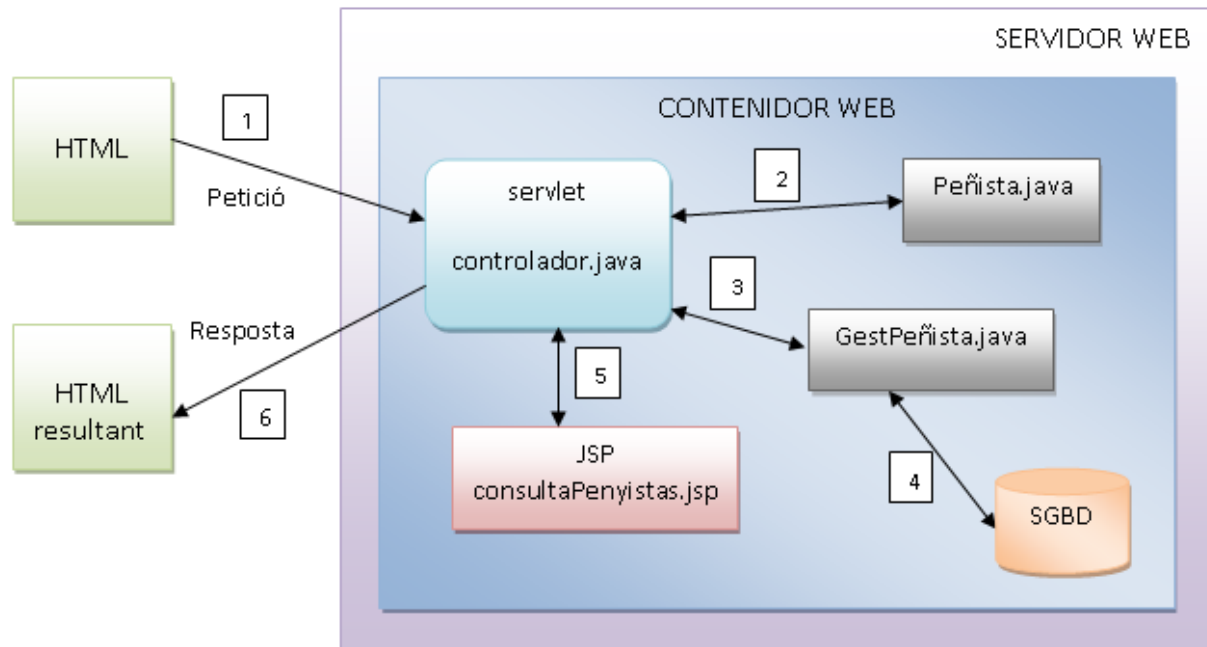


Figura 6.1 Diagrama reduït de consulta de tots els penyistes

- Pas 1: El client utilitza un document HTML per fer una petició al servlet.
- Passos 2 i 3: El servlet, per poder satisfer la petició de l'usuari, utilitza classes java auxiliars (classes POJO). Crida a la classe de negoci 'GestPeñista' perquè aquesta li confeccioni la llista.
- Pas 4: Cal notar com la classe especialitzada en tractar amb informació persistent (mitjançant JPA) és la classe GestPeñista.
- Pas 5: Una vegada que el servlet ja disposa de la llista de penyistes la deixa a la request a disposició de la pàgina JSP encarregada de mostrar els resultats a l'usuari.
- Pas 6: Quan la pàgina JSP es genera s'envia a l'usuari com a pàgina HTML, és a dir, tot el codi ja s'ha convertit en codi estàtic (de presentació)

Generació d'informes

Fem servir la tecnologia JasperReports per mostrar informes en format PDF, excel o word. Per fer això:

- Primer es dissenyen els informes amb l'eina iReports.
- La classe servlet (*Report.java*) és l'encarregada de presentar els informes. Li passem paràmetres i redirigim el nostre servlet *controlador.java* a aquest nou servlet.
- Els informes compilats es troben al directori *WebContent\Informes* del nostre projecte.

Id	Nombre	1º apellido	2º apellido	Fecha nacimiento	NIF/NIE	E-Mail	Fecha alta
1	David	Carazo	admine	01/01/73	admine	admin@europrim.es	01/01/12
2	Pruebas	proves	test	11/01/74	proves	proves@hotmail.es	01/01/12
3	Marcos	Rodriguez	Boes	07/07/92	67543332	marcos@gmail.es	01/08/12
4	Sofia	Perlin		02/12/93	X2134533H	sofia@gmail.com	01/08/12
5	Eduardo	Bombin	Paez	17/03/72	64721291G	eduardo@hotmail.es	01/08/12
6	Marta	Leia	Suller	25/03/76	42348727T	marta@ocm.es	02/08/12
7	Sara	Vilas	Teller	04/12/86	53223143H	sara@terra.com	02/08/12
8	Genis	Falcs	Tomeu	15/06/77	34833029N	genis@gmail.com	02/08/12
9	Pedro	Garcia	Gil	11/02/73	11111111H	pedro@hotmail.es	03/08/12
10	Susana	Marques		30/09/87	X1234567L	simon@hotmail.com	04/08/12
11	Susana	Zaors	Lunger	17/08/85	37238271Y	susana@terra.es	04/08/12
12	Rebeca	Rodriguez	Diaz	22/06/83	38357771G	rebeca@ocm.es	04/08/12
13	Antonio	Estomil	Gracia	09/02/84	35128143J	antonio@gmail.es	05/08/12
14	Alex	Quevedo	Tal	25/06/87	41556241W	alex@hotmail.es	06/08/12
15	Marcos	Lopez	Yafiez	18/09/75	38238273J	marcos@hotmail.com	08/08/12
16	Maria	Lopez	Anias	11/12/84	47213149Z	maria@gmail.com	08/08/12
17	Gloria	Razo	Mager	27/05/73	41827715F	gloria@gmail.com	09/08/12
18	Lucas	Patras	Torcan	01/12/68	29475282T	lucas@terra.com	10/08/12
19	Mercedes	Rodriguez	Tea	13/07/93	47253917A	mercedes@terra.es	10/08/12
20	Juan	Perqual	Hernandez	04/02/83	32145678G	juan@hotmail.es	11/08/12
21	Adolfo	Hisler	Hisler	21/06/74	36552812Q	adolfo@hotmail.es	11/08/12
22	Javier	Toron	Pes	21/06/68	57212149V	javier@ocm.es	12/08/12

Figura 6.2 Captura de pantalla d'un informe en format PDF i excel

Framework de persistència

Copiem les llibreries necessàries per fer servir EclipseLink al buildpath del nostre projecte, donat que Tomcat les necessita en temps d'execució. Mitjançant l'assistent de Eclipse per a projectes basats en JPA creem les classes Entity Beans a partir de les nostres taules de la BD i substituïm el codi JDBC.

Neteja de les JSP's

Hem substituït els scriptlets (codi java) presents a les JSP's de l'aplicació per les etiquetes que ens proporciona la llibreria JSTL, això ens permet una millor ordenació del codi i ens facilitarà el seu manteniment.

Paginació del resultat de les consultes

Quan hem de mostrar els resultats d'una consulta, si el nombre de registres és petit la pàgina obtinguda és agradable i funcional, però si n'hi ha molts la taula resultant és molt llarga, i provoca la insatisfacció de l'usuari a l'haver de desplaçar-se en vertical per consultar totes els dades. La solució que hem emprat és permetre la navegació dels resultats mitjançant el framework de paginació opensource pager-taglib-2.0.

Validación

Validado: adminpe

Peña EuroPrim

CONSULTA DE TODOS LOS PEÑISTAS

Control

Volver

Info.Sesion

Desconectar

Estadísticas

Características

Web segura

Id peñista	Nombre	1er apellido	2do apellido	Fecha nacimiento	NIF/NIE	E-Mail	Fecha alta	Fecha baja	Fondos
2	Pruebas	proves	test	11/01/1974	proves	proves@hotmail.es	01/01/2012	-----	0,00 EUR
3	Marcos	Rodriguez	Boes	07/07/1992	67543332Z	marcos@gmail.es	01/08/2012	-----	103,93 EUR
4	Sofia	Perlin	-----	02/12/1993	X2134533H	sofia@gmail.com	01/08/2012	-----	72,51 EUR
5	Eduardo	Bombin	Paez	17/03/1972	64721291G	eduardo@hotmail.es	01/08/2012	-----	103,93 EUR
6	Marta	Leia	Suller	25/03/1976	42348727T	marta@ozu.es	02/08/2012	-----	125,32 EUR
7	Sara	Viñas	Teller	04/12/1986	53223145H	sara@terra.com	02/08/2012	-----	103,93 EUR
8	Genis	Falcs	Tomeu	15/06/1977	34833029N	genis@gmail.com	02/08/2012	-----	122,52 EUR
9	Pedro	Garcia	Gil	11/02/1973	11111111H	pedro@hotmail.es	03/08/2012	-----	118,47 EUR
10	Simon	Marques	-----	30/09/1967	X1234567L	simon@hotmail.com	04/08/2012	-----	176,43 EUR
11	Susana	Zaos	Lunger	17/08/1985	37238271Y	susana@terra.es	04/08/2012	-----	72,51 EUR
12	Rebeca	Rodriguez	Diaz	22/06/1983	38357771G	rebeca@ozu.es	04/08/2012	16/05/2013	67,34 EUR
13	Antonio	Estomil	Gracia	03/02/1984	35128143J	antonio@gmail.es	05/08/2012	-----	176,43 EUR
14	Alex	Quevedo	Tal	25/06/1987	41556241W	alex@hotmail.es	06/08/2012	-----	70,51 EUR
15	Marcos	Lopez	Yañez	18/09/1975	38238272J	marcos@hotmail.com	08/08/2012	-----	176,43 EUR
16	Maria	Lopez	Arias	11/12/1984	47213149Z	maria@gmail.com	08/08/2012	-----	72,51 EUR
17	Gloria	Razo	Mager	27/05/1973	41827715F	gloria@gmail.com	09/08/2012	-----	103,93 EUR
18	Lucas	Patas	Torcan	01/12/1968	29475282T	lucas@terra.com	10/08/2012	-----	176,43 EUR
19	Mercedes	Rodriguez	Tea	13/07/1993	47253917A	mercedes@terra.es	10/08/2012	-----	72,51 EUR
20	Juan	Perqual	Hernandez	04/02/1983	32145678G	juan@hotmail.es	11/08/2012	-----	103,93 EUR

Figura 6.3 Consulta de penyistes sense framework de paginacio

Validación

Validado: adminpe

Peña EuroPrim

CONSULTA DE TODOS LOS PEÑISTAS

Control

Volver

Info.Sesion

Desconectar

Estadísticas

Características

Web segura

Id peñista	Nombre	1er apellido	2do apellido	Fecha nacimiento	NIF/NIE	E-Mail	Fecha alta	Fecha baja	Fondos
16	Maria	Lopez	Arias	11/12/1984	47213149Z	maria@gmail.com	08/08/2012	-----	72,51 EUR
17	Gloria	Razo	Mager	27/05/1973	41827715F	gloria@gmail.com	09/08/2012	-----	103,93 EUR
18	Lucas	Patas	Torcan	01/12/1968	29475282T	lucas@terra.com	10/08/2012	-----	176,43 EUR
19	Mercedes	Rodriguez	Tea	13/07/1993	47253917A	mercedes@terra.es	10/08/2012	-----	72,51 EUR
20	Juan	Perqual	Hernandez	04/02/1983	32145678G	juan@hotmail.es	11/08/2012	-----	103,93 EUR
21	Adolfo	Hisler	Hisler	21/06/1974	36552812Q	adolfo@hotmail.es	11/08/2012	-----	72,51 EUR
22	Javier	Torron	Pes	21/06/1968	57212149V	javier@ozu.es	12/08/2012	-----	130,52 EUR
23	Nestor	Pontias	Bujas	26/10/1992	44423422A	nestor@lycos.es	13/08/2012	-----	176,43 EUR
24	Andres	Gomez	Gomez	01/01/1981	35234187G	andres@ozu.es	13/08/2012	-----	72,51 EUR
25	Laia	Bastos	Perez	14/06/1982	49123746R	laia@terra.es	14/08/2012	-----	122,52 EUR
26	Roberto	Alcaraz	Peñis	30/09/1988	42874657N	roberto@jazztel.es	14/08/2012	-----	103,93 EUR
27	Luis	Rebollo	Menor	21/10/1994	43215678N	luis@hotmail.es	15/08/2012	-----	72,51 EUR
28	Heinrik	Wolks	-----	28/10/1986	X1872346P	heinrik@hotmail.com	15/08/2012	-----	72,51 EUR
29	ernesto	Segarra	Heras	04/05/1985	42400179r	ernesto@gmail.es	17/08/2012	-----	128,99 EUR
30	Tomas	Aquino	Serz	22/01/1974	26778987H	tomas@gmail.es	17/08/2012	-----	176,43 EUR

16 - 30 de 42
[Primera](#)
|
[Anterior \(15\)](#)
|
[Siguiente \(12\)](#)
|
[Ultima](#)

Créditos: Peña EuroPrim. Copyright © 2013 Proyecto carrera informatica. David Carazo Herrera.

Figura 6.4 Consulta de penyistes amb framework de paginacio

Validacions, Calendari i Captcha

Hem fet servir JQuery ^[g-16] per la validació dels camps dels formularis. Aquesta biblioteca de JavaScript és àmpliament utilitzada per la seva simplicitat per fer una validació simple i la gran quantitat d'opcions que ens permet si volem fer alguna cosa més específica. També ens ha permès reduir la carrega de processament del servidor.

També ens va semblar interessant el fet d'afegir calendaris popup en alguns formularis en comptes d'haver d'escriure directament la data. Hi ha molta varietat d'eines al respecte però nosaltres vam escollir el JSCalendar, que és senzill, pop-up i compatible amb JQuery.

Per afegir més seguretat a l'aplicació, en el formulari d'alta del penyista, hem afegit un Captcha ^[g-17], per evitar que qualsevol programa informàtic, capaç d'omplir de forma automàtica els camps d'un formulari, pugui generar centenars de comptes falses en poc temps i saturar-nos el servidor. El Captcha ha estat creat per nosaltres mateixos, generant un valor hexadecimal afegint-lo a una imatge i després ratllant-lo per sobre.

Validación
Usuario: Anónimo

Control
Volver
Login
Registrarse
Estadísticas

Características
Web segura

Peña EuroPrim
ALTA DE PEÑISTA

Nombre: Introduzca su nombre
1er apellido: Introduzca su 1er apellido
2do apellido: Cognom_2

Fecha de nacimiento: ? Junio, 2013 x
NIF/NIE: 11111111H
Dirección de correo: usuari@uab.c

Contraseña:
Contraseña (confirmar):

Verifica que eres humano, reescribe el texto: b2f41 Introduzca el captcha

Aceptar

Créditos: Peña EuroPrim. Copyright © 2013 Projecte carrera informatica. David Carazo Herrera.

Figura 6.5 Formulari amb validacions, calendari emergent i camp de captcha

(a l'annex-D es poden veure totes les llibreries JavaScript incorporades)

Generació de la Autoritat Certificadora i Web segura (https)

Per fer la nostra aplicació una web segura, hem seguit el següents passos:

- Mitjançant el programa de creació de certificats digitals OpenSSL generació de la nostra pròpia autoritat certificadora (CA)
- Creació del certificat pel servidor (Tomcat)
- Generació de la sol·licitud del certificat del servidor.
- Signatura d'aquesta sol·licitud amb la CA.
- Instal·lació dels certificats de la CA i del signat de Tomcat al magatzem de claus que farà servir el servidor.
- Configuració dels arxius de Tomcat.

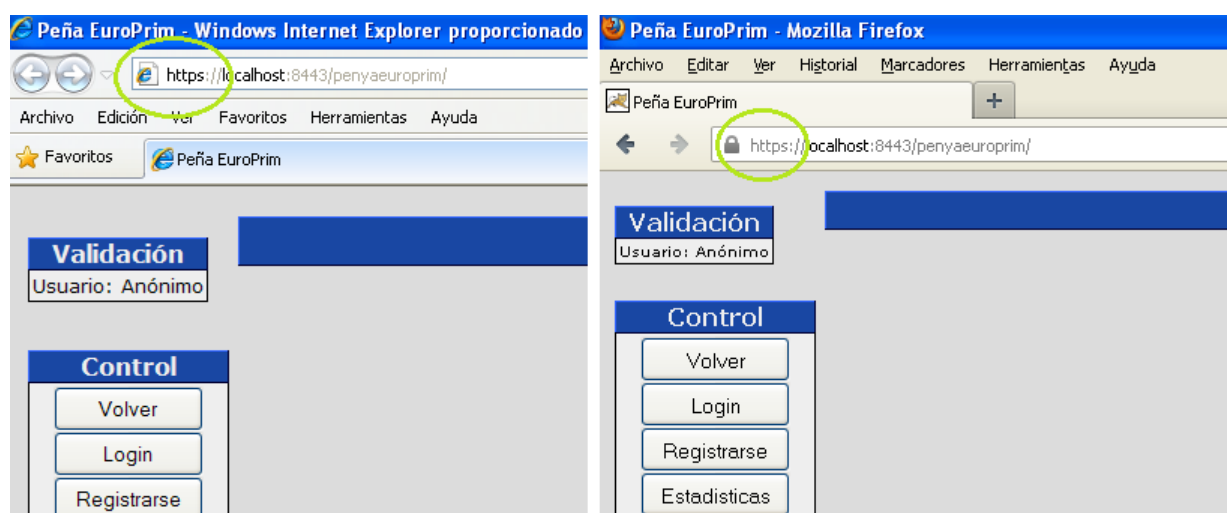


Figura 6.6 Aplicació segura vista a Internet Explorer i Firefox

(a l'*annex-E* es poden veure els passos amb més detall)

Mètode de pagament (penyista i penya)

Pagament per part del penyista a l'hora de participar als sorteigs

Hem afegit a l'aplicació web un botó integrat de PayPal per acceptar pagaments a internet.

Els passos que hem seguits són:

- A través de la plataforma de proves Sandbox, que ofereix PayPal, s'han creat dos comptes (un per la penya i un per un penyista)
- Hem afegit a la JSP corresponent el codi generat per la plataforma Sandbox per insertar un botó de "pagar ahora", i a través de PayPal, tant el penyista com la penya, rebran un justificant del servei.

Figura 6.7 Pàgina de pagament

Pagament per part de la penya a una retirada de fons del penyista

Es comprova que no es puguin retirar més diners dels que hi ha als fons. També es verifica que les dades del compte bancari siguin correctes, mitjançant un algoritme que ho comprova a través dels dígets de control del compte introduït.

Aquest procés generarà una petició en format PDF que serà la que s'haurà d'enviar per correu a l'adreça de la penya.

Validación

Validado: 11111111h

Peña EuroPrim

COBRO DE PREMIOS

Control

Volver

Info Sesión

Desconectar

Estadísticas

Características

Web segura

Para cobrar sus premios, indiquenos la cantidad que desea cobrar, el titular de la cuenta y el número de cuenta donde desea que realicemos la transferencia; luego pulse 'Aceptar' y se generará un informe en formato PDF que tendrá que enviar a la dirección de la Peña (info@penyaeuropim.com). Una vez hayamos recibido el informe procederemos a ingresarle en su nº de cuenta el importe de la cantidad introducida y restaremos de sus fondos dicha cuantía.

La cantidad mínima para poder retirar fondos son 8 EUR, en el supuesto que los fondos no lleguen a dicha cantidad deberá esperar hasta que se alcance esa cifra para que le podamos hacer la transferencia a su cuenta bancaria.

Fondos totales: 118,47 EUR

Importe a retirar:

Titular de la cuenta:

Datos cuenta bancaria: (ej: 12341234121234567890)

Aceptar

Créditos: Peña EuroPrim. Copyright © 2013 Projecte camera informatica. David Carazo Herrera.

Peña EuroPrim

Recuperación de fondos

El peñista con los siguiente datos:

Id peñista: 9

Nombre: Pedro

Apellido: Garcia

Fecha de nacimiento: 11/02/1973

NIEN: 11111111H

e-mail: pedro@hotmail.es

Fecha de alta: 03/08/2012

Notifica la petición de recuperación de la siguiente cantidad de sus fondos en la peña: 105,25 EUR

Acordado que se ingresen en la cuenta bancaria del siguiente titular: Pedro Garcia Gil

Datos de la cuenta bancaria: 12341234161234567890

Una vez la peña haya recibido este informe comprobará que efectivamente haya salido suficiente en los fondos del peñista y acto y seguido procederá a ingresar en la cuenta superior el importe deseado, restando a los fondos totales que dispone en la peña la cantidad que desea percibir en su cuenta para actualizar los fondos del peñista.

IMPORTANTE. El hecho de recibir los informes no implica obligatoriamente que el peñista disponga de dicha cantidad en sus fondos solo es un deseo de retirar de sus fondos un importe que seguramente tendrá que comprobar la administración de la peña si dispone o no de dichos fondos. Esto es así para que la peña esté cubierta legalmente en caso de recibir varios informes malos o contrarios del mismo peñista, el cual repitiendo que no ha llegado al correo con el informe malo a servicio.

Figura 6.8 Pàgina de cobrament de premis i arxiu pdf generat

Framework de presentació

Una aplicació web ha de mostrar el mateix aspecte en totes les pàgines que la conformen i una manera d'aconseguir aquest objectiu és mitjançant l'ús d'un framework de presentació. Hem fet servir SiteMesh, que decora les nostres pàgines JSP en temps d'execució mitjançant filtres servlet per interceptar les crides, decorar-les i presentar-les.

Peña Euromillón-Primitiva

Info. de sesión

Desconectar

Consultas

Seleccione una opción: --consultas--

Consultar

Crear/Seleccionar apuestas peña

Seleccione una opción: --crear/selec--

Crear/Sel

Modificar datos/fondos/peñistas de grupo

Seleccione una opción: --modificaciones--

Modificar

Baja de peñista/apuesta

Seleccione una opción: --bajas--

Baja

Sorteos

Seleccione una opción: --opc. sorteos--

Aceptar

Informes

Seleccione un informe: --informes--

Aceptar

PDF

Excel

Word

Figura 6.9 Menú de l'administrador sense framework de decoració

Peña de jugadores de Primitiva y Euromillón

35

Figura 6.10 Menú de l'administrador amb framework de decoració

Descripció del funcionament general de l'aplicació

La penya compta amb dos grups de penyistes, un que juga als sortejos de la loteria primitiva, i un altre que juga als sortejos l'euromilió.

Cada grup de penyistes juga a les mateixes apostes durant tot el mes, aquest període de temps es considera un *cicle de joc* en la penya.

Les apostes que es juguen durant un *cicle* les marca l'administrador en funció del nombre de penyistes que hi hagi apuntats al grup, és a dir, abans que acabi el mes l'administrador de la penya sabrà quantes persones participen i els diners que hi ha recaptats, amb aquestes dades fa un càlcul de la quantitat d'apostes que es poden realitzar i escull les més propícies d'entre les disponibles, o genera noves apostes que s'inclouran a la BD. Sempre es podran veure les apostes del mes en vigor a través del menú de consultes.

El procés sempre serà el mateix per a totes les participacions (operacions) de sorteig:

- L'administrador crea les participacions de sorteig amb què la penya juga aquella setmana.

- A l'administració de loteries realitza les apostes que corresponen a cada grup (primitiva/euromilió). Un cop s'ha pagat, en el cas de la primitiva, es disposarà dels reintegraments generats per l'administració.
- Les participacions pagades es validaran a través de la secció corresponent del menú (ara només falta esperar a la data dels sortejos)
- Un cop surti el resultat d'un sorteig s'introdueix a la BD. Per dos motius: un perquè quedi com a històric i un altre per relacionar-lo amb la participació que li correspon.
- Quan se sàpiga que és el que percep cada categoria de premis es procedeix a comprovar els encerts de la participació en aquest sorteig.
- Un cop s'ha calculat a quant ascendeix l'import total dels premis d'aquesta participació es passa a emmagatzemar aquest valor, i el programa s'encarrega d'afegir als fons de cada penyista la part proporcional que li toca.

Si un penyista vol jugar amb un grup ha de pagar la quota de *cicle*, però com es paga per avançat haurà d'esperar al mes següent per poder jugar. El pagament s'ha de realitzar abans del dia 20 del mes en vigor, si es passa d'aquesta data, el pagament representarà jugar un mes més tard (p.e. si pagues abans del 20 de gener jugaràs el mes de febrer, però si pagues després jugaràs el mes de març)

Cada vegada que el penyista vulgui participar en un *cicle* haurà de realitzar el pagament, això sí, pagant per avançat. D'aquesta manera li estalviem el destorb d'haver-se de donar d'alta en un grup i cobrar-li cada mes una quota, així només jugarà els *cicles* que ell vulgui; i si es despista un mes no jugarà el següent, però tampoc se li cobrarà; i no hauria de demanar donar-se de baixa si desitgés sortir del grup, ja que no estaria domiciliat cap cobrament.

El penyista disposa d'una secció de peticions, on es generen els informes que necessita per donar-se de baixa de la penya, o per retirar diners del seu fons. Aquests informes són els arxius en format PDF que s'han d'enviar a l'adreça de la penya, perquè, depenent del tipus de petició, es faci finalment efectiva.

7. Conclusions

Moltes vegades sembla que crear una aplicació sota l'arquitectura JEE és molt complicat. Si es segueix l'arquitectura tradicional proposada per Sun s'obté una arquitectura sobredimensionada i molt complexa que no és necessària per al problema que ens interessa resoldre.

La millor arquitectura per a una aplicació és la més simple que compleixi amb els requeriments. Per aquest motiu és molt important tenir clar quins són aquests requeriments. I utilitzar una anàlisi i disseny orientat a objectes permet abstraure'ns en gran part del llenguatge utilitzat i facilita la posterior implementació.

Com hem dit a la planificació temporal, els canvis i les complicacions sorgides en un punt avançat de la fase IV van provocar un desplaçament en el temps de les tasques que anaven a continuació. Encara que com vam realitzar una planificació folgada, això ens va permetre dedicar més hores i poder acabar l'aplicació dins dels límits permesos.

El fet de generar una aplicació segura (https) ens ha permès conèixer una mica més del tema de certificats i la seva signatura per part d'una CA. El motiu de realitzar-lo pel nostre compte era per recrear l'entorn de seguretat i confiança que ofereixen aquest tipus de certificats sense haver de pagar per això, però en una aplicació d'aquest tipus en la realitat el més aconsellable i senzill és comprar-lo a una CA autoritzada.

La confiança la podem entendre com:

“La seguretat que algú té en una altra persona o en alguna cosa. És una qualitat pròpia dels éssers vius, especialment els éssers humans, ja que encara que els animals la posseeixen, aquests ho fan de forma instintiva, al contrari que els humans, que confien conscientment. Com és una cosa que es fa conscient i voluntàriament, suposa treball i esforç aconseguir-la.”

Però en aquest cas en concret la confiança la podem aconseguir pagant un certificat de confiança a una Autoritat Certificadora.

Els Captchas encara que són útils tenen limitacions, impedeixen a persones amb deficiències visuals o auditives accedir als recursos protegits:

(<http://accesibilidadweb.dlsi.ua.es/?menu=que-es-un-captcha-problemas-accesibilidad>)

A més hi ha algunes aproximacions a com es pot trencar un CAPTCHA:

- Usant humans com a mà d'obra barata per reconèixe'ls
- Explotant bugs en la implementació per saltar-se el reconeixement.
- Millorant el programari de reconeixement òptic de caràcters (OCR)

Encara i així, sempre és millor que estigui present a que no hi sigui, per fer una mica més segura l'aplicació.

Els frameworks utilitzats (persistència, paginació, presentació, etc.) també ens ha aportat nous coneixements i ens han facilitat el desenvolupament de l'aplicació.

Un dels punts claus que ens plantejàvem en aquest projecte era: *Cómo resoldre el fet de demostrar que un penyista, amb les dades que havia aportat, realment existia*, ja que ens podriem trobar amb un problema legal a l'hora de verificar que els nostres penyistes són persones reals, i la penya no és sino una tapadora pel blanqueig de diners, per exemple.

La conclusió a la qual vam arribar és que una targeta està associada a un compte bancari, i per obrir un compte bancari s'ha de ser major d'edat i complir una sèrie de requisits específics, de manera que, si un penyista paga amb targeta estem demostrant que és una persona real identificada per una entitat bancària.

La solució que hem trobat és fer pagar al penyista a través de PayPal, on es permet pagar amb un compte de PayPal o amb tarjeta. El resultat final és el mateix perquè quan algú es dona d'alta a PayPal ha d'afegir un compte bancari que es vincularà amb el compte d'usuari. O sigui que d'una manera o d'una altra tenim a una persona que pagarà amb tarjeta.

Altre punt clau era : *Demostrar que en el moment d'abonar fons al penyista, aquests diners van a parar a una persona real i que representa a aquest penyista.* En aquest cas el que hem fet, igual que fan moltes webs d'apostes consultades, és demanar al penyista un nro. de compte bancari per fer la transferència d'aquests fons, on el titular d'aquest compte ha de ser el penyista en qüestió. A la transferència s'haurà de reflexar, a conceptes de l'ingrés, que és un pagament de premis. Altra opció possible seria ingressar a final del mes els fons totals del penyista a la seva compte corrent. Com ja hem dit abans, si hi ha un compte bancari associat a aquesta persona, llavors existeix i no hi ha cap il·legalitat.

Durant la generació de l'aplicació, també he trobat força interessant un aspecte que molta gent passa per alt quan es dona d'alta en pàgines web o qualsevol tipus de servei: Llegir la **política de privacitat o condicions generals**. És important que abans de donar-se d'alta a qualsevol lloc es faci aquest pas, perquè en el moment que es doni d'alta també es demanarà acceptar aquestes condicions, i si no es té clar el que hi posa pot trobar-se després amb una sorpresa.

L'aplicació s'ha provat als tres navegadors més importants: *Internet Explorer 8*, *Mozilla Firefox 21.0* i *Google Chrome 27.0*, i podem dir que és funcional. Com a web segura ha funcionat bé a IE i Firefox; mentre que amb Chrome, com el nostre servidor està en local, no s'ha pogut validar la seva identitat amb una entitat emissora externa, encara que accepta el nostre certificat signat i la connexió està xifrada. Però com hem citat abans això no passaria a una aplicació real ja que comprariem el certificat a una CA.

De manera que en línies generals hem assolit els nostres objectius i millorat els nostres coneixements.

Finalment, el temps dedicat a la realització de la memòria ha sigut de 144 hores.

Línies de continuació

Si analitzem el nostre model veiem que hi ha un conjunt de tasques repetitives: rebre els paràmetres d'entrada de les peticions i fer validacions, trucar a la lògica de negoci

i escollir la vista a mostrar. Aquestes tasques es podrien implementar amb un framework per a la capa de presentació com: Struts, Java Server Faces, Spring (Spring MVC),...I el seu ús en aquesta capa ens oferiria avantatges com:

- Desacoblar la capa de presentació de la capa de negoci en components separats.
- Simplificar i estandarditzar la validació dels paràmetres d'entrada.
- Permetre un nivell molt alt de reutilització.
- Simplificar moltes tasques repetitives.

Altres millores a fer serien:

- Afegir el patró de disseny Facade a la capa del model de dades per separar els components de la Vista i el Controlador dels components del Model.
- Internacionalitzar l'aplicació (i18n).
- Enviament d'e-mails desde l'aplicació.
- Afegir logs per facilitar la comprovació del sistema, la depuració, el rastreig d'errors.

8. Annexos

A. Glossari

[g-1] Autoritat Certificadora (CA)

L'autoritat de certificació verifica la identitat del sol·licitant d'un certificat abans de la seva expedició o, en cas de certificats expedits amb la condició de revocats, elimina la revocació dels certificats en comprovar aquesta identitat. Els certificats són documents que recullen certes dades del seu titular i la seva clau pública i estan signats electrònicament per l'Autoritat de Certificació amb la clau privada.

Un certificat de CA pot estar auto-signat quan no hi ha cap CA de rang superior que el signi. Aquest és el cas dels certificats de CA arrel, l'element inicial de qualsevol jerarquia de certificació.

Si està instal·lada una CA en el repositori de CA de confiança de cada navegador, qualsevol certificat signat per aquesta CA es podrà validar, ja que es disposa de la clau pública amb la qual verificar la signatura que porta el certificat.

[g-2] DOM (Document Object Model)

El DOM és la interfície que permet accedir i manipular, mitjançant la programació, els continguts d'una pàgina web. Proporciona un mecanisme estàndard per accedir des de codi a la informació continguda en un document de text basat en etiquetes (marques)

Mitjançant DOM es poden manipular documents XML o XHTML com un conjunt d'objectes organitzats de forma jeràrquica, on tots els elements són nodes.

[g-3] Fulles d'estil (CSS)

Les fulles d'estil són la innovació més important per enriquir un document estructurat escrit en HTML (i per extensió en XHTML). Aquestes fulles ens permeten especificar l'estil d'una pàgina HTML (marges, espais, tipus de lletra, colors, etc) de forma independent a la especificada en l'estructura de la pàgina (paràgrafs, capçaleres, títols, enllaços, etc)

[g-4] Servlets

Un servlet és un objecte Java que estén la funcionalitat d'un servidor web, rep peticions HTTP i genera contingut dinàmic com a resposta a aquestes peticions.

Els servlets "viuen" dins d'un contenidor web i es carreguen i executen dinàmicament com a resposta a les peticions que els clients fan a una URL determinada. El cicle de vida dels servlets ho controla el contenidor on s'ha desplegat.

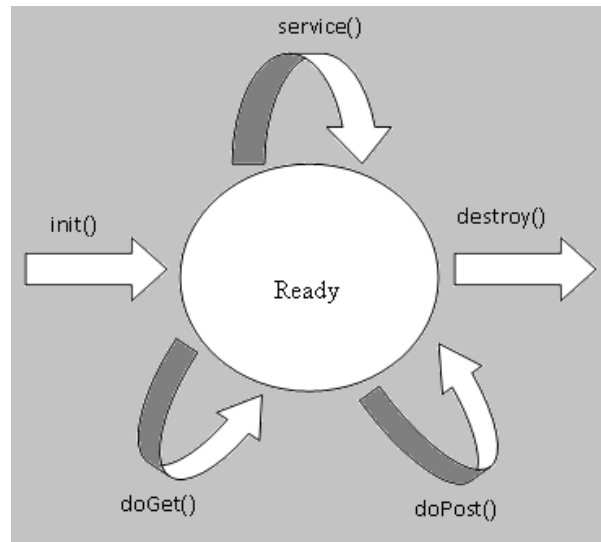


Figura g-4.1. Cicle de vida d'un servlet

Els servlets basen el seu funcionament en el paradigma de petició-resposta sota el protocol HTTP. Les peticions HTTP més habituals són peticions POST o GET. Els servlets tenen els mètodes *doPost()* i *doGet()* per servir directament aquestes peticions. Si el programador ha sobreescrit el mètode *doPost()* i arriba una petició POST s'executa el codi que hi hagi en aquest mètode. Ídem per *doGet()*.

Quan el contenidor rep una petició que està mapejada amb un servlet es produeixen els següents passos:

1. El contenidor crea els objectes que modelen la petició i la resposta (*HttpServletRequest* / *HttpServletResponse*)
2. Si no hi ha cap instància del component en memòria, el contenidor crea un i l'inicialitza cridant al mètode *init()* i després al mètode *service()*.
3. Si ja hi havia una instància del component, crida al mètode *service()* directament.

El desenvolupador programa el comportament del servlet per a cada petició que es faci al mètode *service()*: *doPost()*, *doGet()*, *doDelete()*, *doHead()*, etc.

Els servlets no s'executen com un procés separat, i ens evitem de crear un procés nou a cada petició. Com que una sola instància del servlet s'encarrega de respondre a totes les peticions concurrentment, també ens permet estalviar memòria i gestionar fàcilment dades persistents.

[g-5] JSP (Java Server Pages)

La tecnologia Java Server Pages (JSP) és una especificació oberta i gratuïta desenvolupada per Sun Microsystems com a alternativa a Active Server Pages (ASP) de Microsoft. A més, JSP ens permet separar el disseny del contingut, amb totes les avantatges que això suposa.

Una pàgina JSP és un document similar a una pàgina HTML però que a més de generar codi estàtic pot incloure elements per generar contingut dinàmic en resposta a una petició HTTP.

Els elements per generar contingut dinàmic són fragments de codi escrits en java (scriptlets) i unes etiquetes que interactuen amb objectes java del servidor (<jsp:useBean...>, etc).

La tecnologia de servlets i la de JSP estan molt relacionades, de fet, les JSP són transformades internament a servlets quan es despleguen en el contenidor web, abans de servir les peticions als clients.

Per executar les pàgines JSP, necessitem que el servidor web tingui un contenidor web que compleixi les especificacions de JSP i de servlet. Aquest contenidor manega l'execució de les pàgines JSP i dels servlets que s'executen en el servidor.

[g-6] EJB (Enterprise Java Beans)

Els EJB's són, segons Sun, la tecnologia més important de l'especificació de JEE. Es tracta d'una tecnologia de components de servidor que permet el desenvolupament i el desplegat d'aplicacions empresarials distribuïdes basades en components.

Les aplicacions que desenvolupem amb EJB's són escalables, portables, segures, transaccionals, multiusuari i distribuïdes. Aquestes característiques els proporciona un grau molt alt de reusabilitat, i permet que el contenidor implementi els serveis de baix nivell i els desenvolupadors es puguin centrar en la lògica de negoci.

Els EJB són components desenvolupats amb Java que compleixen unes especificacions i, per tant, es poden desplegar en qualsevol contenidor de EJB's d'un

servidor d'aplicacions compatible amb la plataforma JEE: Glassfish, JBoss, Geronimo, WebSphere, etc.

L'especificació de EJB defineix tres tipus bàsics de EJB:

- EJB's de sessió: Modelen processos de negoci. En aquest tipus de components es realitza la implementació dels mètodes de negoci.
- EJB's d'entitat: Modelen dades de negoci. Aquests components permeten representar en forma d'objectes java dades de taules de bases de dades relacionals. Representen les entitats del domini.
- EJB's de missatge: Solen modelar processos de negoci com els EJB de sessió, però són components sense estat, de servidor, transaccionals que serveixen per rebre missatges asíncrons d'altres sistemes.

Per a la construcció d'un EJB és necessari:

- Bean: És la classe (POJO) que implementa el component. En el cas dels EJB d'entitat contindrà els atributs i mètodes associats al model de dades utilitzant JPA.
- Interfície de negoci (POJI): Definició dels mètodes exposats pel bean.
- Descriptor de desplegament (Deployment descriptor): Fitxer XML amb informació necessària perquè el servidor d'aplicacions sàpiga com desplegar el component.

[g-7] SSL (Secure Sockets Layer)

SSL proporciona autenticació i privacitat de la informació entre extrems sobre Internet mitjançant l'ús de criptografia. Habitualment, només el servidor és autenticat (es garanteix la seva identitat) mentre que el client es manté sense autenticar.

SSL implica una sèrie de fases bàsiques:

- Negociar entre les parts l'algorisme que s'utilitzarà en la comunicació:
 - Per criptografia de clau pública: RSA, Diffie-Hellman, DSA o Fortezza;
 - Per xifrat simètric: RC2, RC4, IDEA, DES, Triple DES i AES;
 - Amb funcions hash: MD5 o de la família SHA.

- Intercanvi de claus públiques i autenticació basada en certificats digitals.
- Xifrat del trànsit basat en xifrat simètric.

SSL s'executa en una capa entre els protocols d'aplicació com HTTP, SMTP, NNTP i sobre el protocol de transport TCP, que forma part de la família de protocols TCP/IP. S'utilitza en la majoria dels casos al costat de HTTP per formar HTTPS. HTTPS és usat per assegurar pàgines *www* per a aplicacions de comerç electrònic, utilitzant certificats de clau pública per verificar la identitat dels extrems.

[g-8] Eclipse

Hi ha diferents programes comercials que permeten desenvolupar codi Java. Diverses companyies com Sun, IBM, etc. Distribueixen gratuïtament el Java Development Kit (JDK). Es tracta d'un conjunt de programes i llibreries que permeten desenvolupar, compilar i executar programes en Java. Existeix també una versió reduïda del JDK, anomenada JRE (Java Runtime Environment) destinada únicament a executar codi Java (no permet compilar).

Els IDEs (Integrated Development Environment), són entorns de desenvolupament integrats. En un mateix programa és possible escriure codi i executar-lo sense haver de canviar d'aplicació. Alguns inclouen una eina per realitzar debug gràficament, enfront de la versió que incorpora el JDK basada en la utilització d'una consola bastant difícil i pesada d'utilitzar. Aquests entorns integrats permeten desenvolupar les aplicacions de forma molt més ràpida, incorporant en molts casos llibreries amb components ja desenvolupats, els quals s'incorporen al projecte o programa.

En el mateix entorn, disposem d'un depurador i d'una consola de missatges que ens ajudaran en el desenvolupament dels diferents mòduls.

Una de les grans avantatges d'Eclipse és que a l'estar desenvolupat en Java és multiplataforma. Eclipse va ser desenvolupat originàriament per IBM. Actualment és desenvolupat per la Fundació Eclipse, una organització independent sense ànim de lucre que fomenta una comunitat de codi obert i un conjunt de productes complementaris, capacitats i serveis.

[g-9] Tomcat

Tomcat (també anomenat Jakarta Tomcat o Apache Tomcat) funciona com un contenidor de servlets desenvolupat sota el projecte Jakarta a l'Apache Software Foundation. Tomcat implementa les especificacions dels servlets i les JavaServer Pages (JSP) de Sun Microsystems.

Tomcat és un servidor web amb suport de servlets i JSP's. Tomcat no és un servidor d'aplicacions. Inclou el compilador Jasper, que compila JSP's convertint-les en servlets.

Tomcat pot funcionar com a servidor web per si mateix. En els seus inicis va existir la percepció que l'ús de Tomcat de forma autònoma era només recomanable per a entorns de desenvolupament i entorns amb requisits mínims de velocitat i gestió de transaccions. Avui dia ja no existeix aquesta percepció i Tomcat és usat com servidor web autònom en entorns amb alt nivell de tràfic i alta disponibilitat. Atès que Tomcat va ser escrit en Java, funciona en qualsevol sistema operatiu que disposi de la màquina virtual Java.

[g-10] MySQL

És un sistema gestor de base de dades relacional, multi-fil i multi-usuari. MySQL és un software lliure amb un esquema de llicenciament dual. D'una banda s'ofereix sota la GNU GPL per a qualsevol ús compatible amb aquesta llicència, però per a aquelles empreses que vulguin incorporar-lo en productes privatis han de comprar a l'empresa una llicència específica que els permeti aquest ús.

MySQL és utilitzat en molts desenvolupaments web i actualment té milions d'usuaris arreu del món. Permet instal·lar-se en la majoria de sistemes operatius, i entre les seves característiques principals destaquem la velocitat en les operacions i la seguretat referent a la connectivitat. Amb aplicacions web hi ha baixa concurrència en la modificació de dades i en canvi l'entorn és intensiu en lectura de dades, el que fa a MySQL ideal per a aquest tipus d'aplicacions.

[g-11] Classes POJO (Plain Old Java Object)

Els POJO són classes java normals, amb el gran avantatge de ser objectes molt lleugers que no afegeixen cap tipus de càrrega addicional (gestió de transaccions, seguretat, cicle de vida, etc)

Són molt fàcils de desenvolupar, a més de proporcionar molt bon rendiment. El seu principal inconvenient és que no tenim cap de les característiques que tenen els EJB's i si les necessitem les haurem de implementar (o utilitzar algun framework que les proporcioni)

[g-12] ORM (Object Relational Mapping)

És una tècnica de programació que ens permet vincular els objectes usats en el nostre model de l'aplicació amb una base de dades relacional.

El principal problema sorgeix perquè avui en dia, pràcticament totes les aplicacions estan dissenyades per utilitzar l'Orientació a Objectes, mentre que les bases de dades més esteses són del tipus relacional.

Les bases de dades relacionals només permeten guardar tipus de dades primitius (sencers, cadenes de text, etc) pel que no es pot guardar de forma directa els objectes de l'aplicació en les taules, sinó que aquests s'han de convertir abans en registres, que en general afecten diverses taules. En el moment de tornar a recuperar les dades, cal fer el procés contrari, s'han de convertir els registres en objectes.

És llavors quan ORM cobra importància, ja que s'encarrega de forma automàtica de convertir els objectes en registres i viceversa, simulant així tenir una base de dades orientada a objectes.

Avantatges de fer servir ORM:

- Rapidesa de desenvolupament: Tu crees la BD i l'eina automàticament llegeix l'esquema de taules i relacions i crea un model ajustat. Això treu molt codi repetitiu.

- Abstracció del motor de base de dades: El sistema ORM genera automàticament les consultes a la BD per convertir els registres en objectes (i viceversa) i aquestes han de poder adaptar-se als diferents proveïdors (MySQL, Oracle, PostgreSQL, etc)
- Llenguatge propi per a consultes a la base de dades: Permetent a la gent oblidar-se de la sintaxi SQL per utilitzar el propi llenguatge de l'eina.

[g-13] JSTL vs Scriptlets

Quan s'insereix Java en HTML parlem de scriptlets. Els scriptlets han estat la solució tradicional per a la creació de pàgines JSP des dels principis d'aquesta tecnologia, però amb el temps es va veure que no resultava molt còmode barrejar HTML i Java, pel que van aparèixer les anomenades llibreries d'etiquetes, que sense ser la solució definitiva permeten crear pàgines JSP d'una manera més polida.

Què és JSTL? (JSP Standard Tag Library)

JSTL és un conjunt de llibreries d'etiquetes simples i estàndards que encapsulen la funcionalitat principal que és usada comunament per escriure pàgines JSP.

Les etiquetes JSTL estan organitzades en 4 llibreries:

- *core*: Comprèn les funcions script bàsiques com loops, condicionals, i entrada/sortida.
- *xml*: Comprèn el processament de XML.
- *fmt*: Comprèn la internacionalització i format de valors com de moneda i dates.
- *sql*: Comprèn l'accés a bases de dades.

Quin és el problema amb els scriptlets JSP?

Exemple d'un scriptlet JSP: `<% int comptador = 100;%>`

L'especificació JSP és en una tecnologia estàndard per a la creació de llocs web dinàmics en Java, i el problema és que han aparegut algunes debilitats:

- El codi Java embegut en scriptlets és desordenat.
- Un programador que no coneix Java no pot modificar el codi Java encastat, anul·lant un dels majors beneficis dels JSP: permetre als dissenyadors i persones que escriuen la lògica de presentació que actualitzin el contingut de la pàgina.
- El codi de Java dins l'scriptlets JSP no pot ser reutilitzat per altres JSP, per tant la lògica comuna acaba sent re-implementada en múltiples pàgines.
- La recuperació d'objectes fora de l'HTTP Request i Session és complicada. Cal fer el càsting d'objectes i això fa que haguem de importar més classes en els JSP.

Com millora aquesta situació la llibreria JSTL?

- Com que les etiquetes JSTL són XML, aquestes etiquetes s'integren neta i uniformement a les etiquetes HTML.
- Les 4 llibreries d'etiquetes JSTL inclouen la majoria de les funcionalitats que seran necessàries a una pàgina JSP.
- Les etiquetes JSTL poden referenciar objectes que es trobin en els ambients Request i Session sense conèixer el tipus de l'objecte i sense necessitat de fer el càsting.
- Els JSP EL (Expression Language) faciliten les trucades als mètodes get i set en els objectes Java.

[g-14] JPA (Java Persistence API)

Java Persistence API, més coneguda per JPA, és l'API de persistència desenvolupada per a la plataforma Java EE i inclosa en l'estàndard EJB3. És un model de EJB d'entitat basat en components JavaBean (POJO)

Amb JPA es poden fer coses com:

- Escriure el codi d'una classe i encara que no hi hagi la taula corresponent a la base de dades JPA s'encarregarà de crear-la a partir de la classe. Si la taula ja existeix, llavors la utilitza.
- És igual el SGBD que utilitzem, es podria, per exemple, escriure l'aplicació per accedir a MySQL i després amb un lleuger canvi accedir a Oracle o qualsevol altre.
- Sincronització. Si volem afegir dues columnes noves a certa taula quan portem escrites diverses sentències SQL ens tocava canviar les sentències. Amb JPA és tan senzill com anar al codi de la nostra classe java i afegir dos atributs nous que es mapejaren com camps nous de la taula.
- Es pot també mantenir la persistència no necessàriament mitjançant una base de dades, potser interressi emmagatzemar un objecte a un fitxer de text pla o XML.

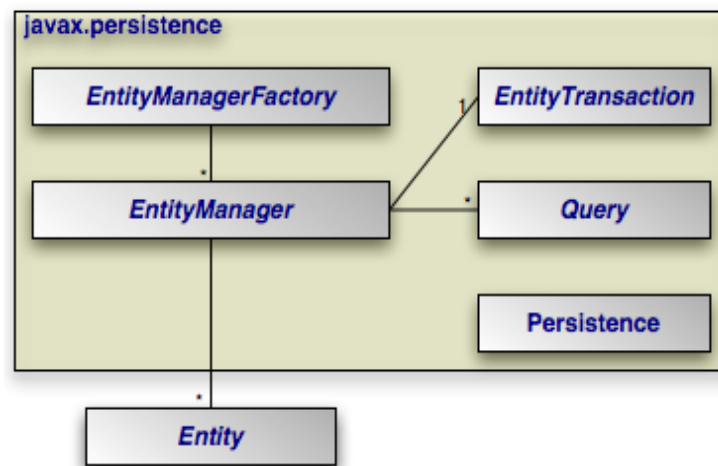


Figura g-14.1 Relació els components principals de l'arquitectura JPA

EntityManagerFactory: Nos ayuda a crear objetos de EntityManager utilizando el patrón de diseño Factory (fábrica)

EntityManager: Es la interfaz principal de JPA utilizada para la persistencia de las aplicaciones. Cada EntityManager puede realizar operaciones CRUD (Create, Read, Update, Delete) sobre un conjunto de objetos persistentes.

Entity: Es una anotación java que se coloca a nivel de clases java serializables y que cada objeto de una de estas clases anotadas representa un registro de una BD.

EntityTransaction: Permite operaciones sobre datos persistentes de manera que agrupados formen una unidad de trabajo transaccional, en el que todo el grupo sincroniza su estado de persistencia en la BD o todos fallan en el intento, para mantener la integridad de los datos.

Query: Està implementada per cada proveïdor de JPA per trobar objectes persistents fent servir cert criteri de cerca.

Persistence: Conté mètodes estàtics d'ajuda per obtenir una instància de EntityManagerFactory d'una forma independent al proveïdor de la implementació de JPA.

JPA estandarditza el suport per a consultes utilitzant JPQL (Java Persistence Query Language) i SQL.

[g-15] JDBC (Java Database Connectivity)

JDBC és una biblioteca de classes que permet la connexió amb una base de dades utilitzant Java. Permet realitzar operacions (consultes, actualitzacions, etc) sobre bases de dades relacionals utilitzant SQL (Structured Query Language)

JDBC està compost per dues capes:

- L'API JDBC, el qual s'encarrega de comunicar amb l'API de gestor de drivers JDBC enviant les sentències SQL
- Un gestor de drivers JDBC que comunica (de forma transparent per al programador) amb els diferents controladors disponibles per a la base de dades.

L'habitual és que els fabricants de SGBD subministrin controladors JDBC de tipus 3 o 4. En el nostre cas tractarem amb el tipus 4, (d'extrem a extrem és java) .Aquest tipus, el JDBC natiu, converteix les crides JDBC a crides directes al protocol natiu de

xarxa del SGBD sense una capa de software intermèdia. Aquest driver s'executarà al servidor.

Els passos a realitzar per utilitzar JDBC en una aplicació són els següents:

1. Enregistrar el driver pel SGBD (en aquest cas per MySQL) :
`Class.forName("com.mysql.jdbc.Driver").newInstance();`
2. Construir una URL a la base de dades (per exemple):
`jdbc:mysql://localhost:3306/dbpenya?user=usuari&password=p_usuari`
3. Establir una connexió:
`Connection conn=DriverManager.getConnection(URL);`
4. Execució de sentències SQL (per exemple una consulta):
`Statement st = conn.createStatement();`
`ResultSet rs = st.executeQuery("SELECT * FROM penyistas;");`
5. Obtenció dels resultats:
`while (rs.next()) {`
`String nom = rs.getString("nom");`
`...;`
`}`
6. Tancament de la connexió (una connexió consumeix molts recursos tant a nivell de java com a nivell de la base de dades):
`conBD.close();`

[g-16] JQuery

JQuery és una biblioteca de JavaScript que facilita la manera d'interactuar amb els documents HTML, manipular l'arbre DOM, gestionar esdeveniments, desenvolupar animacions i afegir interacció amb la tecnologia AJAX a pàgines web.

jQuery és un software lliure i de codi obert, posseeix un doble llicenciament sota la Llicència MIT i la Llicència pública general de GNU v2, permetent el seu ús en projectes lliures i privatis. Més informació a <http://api.jquery.com>

[g-17] CAPTCHA

Captcha són les sigles de Completely Automated Public Turing test to tell Computers and Humans Apart (Prova de Turing pública i automàtica per diferenciar màquines i humans)

Es tracta d'una prova desafiament-resposta utilitzada en computació per determinar quan l'usuari és o no humà. Consisteix en que l'usuari introdueixi correctament un conjunt de caràcters que es mostren en una imatge distorsionada que apareix en pantalla. Se suposa que una màquina no és capaç de comprendre i introduir la seqüència de forma correcta, i que només l'humà podria fer-ho.

Els captchas s'utilitzen per a evitar que robots, també anomenats spambots, puguin utilitzar certs serveis. Per exemple, perquè no puguin participar en enquestes o fòrums de discussió, registrar-se per utilitzar comptes de correu i més recentment, per evitar que un bot d'aquest tipus pugui enviar correu brossa (el remitent ha de passar el test abans que es lliuri al destinatari)

B. Descripció textual dels casos d'ús

Cas d'ús	Iniciar sessió
Actors	Usuari
Objectiu	Accedir al menú del penyista o de l'administrador.
Precondició	L'usuari encara no està validat, si ja s'està logat aquest botó no surt.
Postcondició	L'usuari estarà autenticat.
Escenari	<ol style="list-style-type: none">1. L'actor introdueix el nom d'usuari i contrasenya.2. Pressiona el botó Aceptar3. L'aplicació carrega el menú corresponent depenent del tipus d'usuari que sigui, "Penyista" o "Administrador".
Flux alternatiu	Si l'actor no existeix al sistema, està donat de baixa o la contrasenya és incorrecta es mostra un error.

Cas d'ús	Donar-se d'alta
Actors	Usuari
Objectiu	Formar part de la penya per poder participar en els sortejos.
Precondició	És un usuari no validat qui fa el registre, si ja s'està logat aquest botó no surt.
Postcondició	Es crea un nou penyista a la taula penyistas.

Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix el seu nom, cognoms, data de naixement, NIF / NIE, adreça de correu electrònic i contrasenya. 2. Pressiona el botó Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si tot és correcte s'enregistra el nou penyista a la BD. 5. Surt missatge de confirmació.
Flux alternatiu:	Si el penyista ja existeix, està donat de baixa o és menor d'edat es mostra un error.

Cas d'ús	Veure estadístiques
Actors	Usuari, Peñista i Administrador
Objectiu	Veure les dades i gràfiques referents a la penya de sortejos.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. L'actor pressiona el botó Estadísticas 2. El sistema recupera dades de la BD i genera uns gràfics de guanys/inversió dels diferents grups i una taula amb una sèrie de dades referents a la penya.
Flux alternatiu:	Cap

Cas d'ús	Tornar
Actors	Usuari, Peñista i Administrador
Objectiu	En el cas d'Usuari es tornarà al menú inicial de l'aplicació, i en la resta de casos es tornarà al menú principal corresponent al tipus d'actor que sigui.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. L'actor pressiona el botó Volver 2. El sistema comprova el tipus d'actor per adreçar-lo al seu menú corresponent.
Flux alternatiu:	Si la sessió de l'usuari validat a caducat s'envia al menu inicial de l'aplicació.

Cas d'ús	Informació de sessió
Actors	Peñista i Administrador
Objectiu	Veure les dades de la sessió actual.
Precondició	És un usuari validat qui fa l'acció, si no s'està logat aquest botó no surt.
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. L'actor pressiona el botó Info. Sesion 2. Surten les dades de sessió
Flux alternatiu:	Cap

Cas d'ús	Tancar sessió
Actors	Peñista i Administrador
Objectiu	Per poder sortir de l'aplicació i tancar la sessió actual.

Precondició	És un usuari validat qui fa l'acció, si no s'està logat aquest botó no surt.
Postcondició	Es tanca la sessió actual.
Escenari:	1. L'actor pressiona el botó Desconnectar 2. Confirmació de l'acció. 3. Surt missatge de confirmació.
Flux alternatiu:	Si es cancel.la l'acció continuem dins el menu.

Cas d'ús	Consultes
Actors	Peñista i Administrador
Objectiu	Seleccionar una opció per veure la taula amb les dadesdes.
Precondició	L'actor ha d'estar validat.
Postcondició	Cap
Escenari:	1. L'actor selecciona la consulta desitjada. 2. Pressiona el botó Consultar 3. Anem a l'opció escollida.
Flux alternatiu:	Cap

Cas d'ús	Consultar apostes grup primitiva
Actors	Peñista i Administrador
Objectiu	Veure totes les apostes que juguen el grup de penyistes de la primitiva el més actual.
Precondició	Cap
Postcondició	Cap
Escenari:	1. El sistema accedeix a la BD per recuperar les apostes que s'hi juguen actualment als sortejos de la primitiva. 2. Treu per pantalla una taula amb les dades resultants (id. aposta,números combinació)
Flux alternatiu:	Si no hi ha cap aposta la taula surt buida.

Cas d'ús	Consultar apostes grup euromilió
Actors	Peñista i Administrador
Objectiu	Veure totes les apostes que juguen el grup de penyistes de l'euromilió el més actual.
Precondició	Cap
Postcondició	Cap
Escenari:	1. El sistema accedeix a la BD per recuperar les apostes que s'hi juguen actualment als sortejos de l'euromilió. 2. Treu per pantalla una taula amb les dades resultants (id. aposta,números combinació i estrelles)
Flux alternatiu:	Si no hi ha cap aposta la taula surt buida.

Cas d'ús	Consultar participacions
Actors	Peñista
Objectiu	Veure totes les participacions del penyista (les jugades i les que actualment hi juga).
Precondició	Cap

Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar totes les participacions del penyista. 2. Treu per pantalla una taula amb les dades resultants (id. participació, tipus d'aposta, data sorteig, reintegraments en cas d'aposta primitiva, premis i id. sorteig)
Flux alternatiu:	Si no hi ha cap aposta la taula surt buida.

Cas d'ús	Modificacions
Actors	Peñista i Administrador
Objectiu	Seleccionar una opció de les disponibles a modificar segons el perfil d'actor.
Precondició	L'actor ha d'estar validat.
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. L'actor selecciona una opció. 2. Pressiona el botó Modificar 3. Anem a l'opció escollida.
Flux alternatiu:	Cap

Cas d'ús	Modificar dades personals
Actors	Peñista
Objectiu	Permetre a l'actor canviar l'adreça de correu.
Precondició	Cap
Postcondició	S'actualitza la taula de penyistas amb les noves dades.
Escenari:	<ol style="list-style-type: none"> 1. L'actor canvia les dades del camp e-mail. 2. Pressiona el botó Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si tot està bé surt missatge de confirmació.
Flux alternatiu:	Cap

Cas d'ús	Modificar contrasenya
Actors	Peñista i Administrador
Objectiu	Permetre a l'actor canviar el seu password.
Precondició	Cap
Postcondició	S'actualitza la taula penyistas amb les noves dades.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix la contrasenya antiga, la nova i torna a escriure la nova. 2. Pressiona el botó Aceptar 3. En cas d'error es demana corregir els camps que no estiguin bé. 4. Si no hi ha cap error surt missatge de confirmació.
Flux alternatiu:	Cap

Cas d'ús	Sortejos
Actors	Peñista i Administrador
Objectiu	Seleccionar una opció de les que hi ha disponibles a la secció de sortejos segons el perfil d'actor.
Precondició	L'actor ha d'estar validat.

Postcondició	Cap
Escenari:	1. L'actor selecciona una opció. 2. Pressiona el botó Aceptar 3. Anem a l'opció escollida.
Flux alternatiu:	Cap

Cas d'ús	Participar en grup
Actors	Peñista
Objectiu	Participar en el grup o grups de la penya que hi juguen als sorteigs de la primitiva i l'euromilió.
Precondició	Cap
Postcondició	Cap
Escenari:	1. L'actor escull l'opció de pagament. 2. Pressiona el botó Pagar ahora 3. S'envia a l'actor a PayPal. 4. Un cop s'ha acabat el procés, PayPal retorna l'actor a l'aplicació.
Flux alternatiu:	Cap

Cas d'ús	Comprovar resultats participació
Actors	Peñista i Administrador
Objectiu	Comprovar els encerts d'una participació determinada.
Precondició	Si l'actor és el penyista, aquest ha d'estar dins del grup de penyistes que juguen en la participació, la participació ha d'estar validada i tenir un sorteig associat.
Postcondició	Cap
Escenari:	1. L'actor introdueix el id. de participació (operació) 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors mostra les dades de la participació. 5. Pressiona el botó Comprobar 6. Surten per pantalla els encerts.
Flux alternatiu:	Cap

Cas d'ús	Peticions
Actors	Peñista
Objectiu	Seleccionar una de les opcions possibles per generar l'informe necessari.
Precondició	L'actor ha d'estar validat.
Postcondició	Cap
Escenari:	1. L'actor selecciona la petició desitjada. 2. Pressiona el botó Aceptar 3. Anem a l'opció escollida.
Flux alternatiu:	Cap

Cas d'ús	Baixa de la penya
Actors	Peñista
Objectiu	Informar a la penya que l'actor es vol donar de la baixa.

Precondició	Cap
Postcondició	Es genera l'informe de baixa en format PDF que s'haurà d'enviar a l'e-mail de la penya.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix el titular del compte i el nº de compte bancari. 2. Pressiona el botó Aceptar 3. Si hi ha errors es demana la seva correcció.
Flux alternatiu:	Cap

Cas d'ús	Recuperació de fons
Actors	Peñista
Objectiu	Informar a la penya que l'actor vol treure diners del seu fons.
Precondició	L'actor ha de tindre fons suficients.
Postcondició	Es genera l'informe de recuperació de fons en format PDF que s'haurà d'enviar a l'e-mail de la penya.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix la quantitat, el titular del compte i el nº de compte bancari. 2. Pressiona el botó Aceptar 3. Si hi ha errors es demana la seva correcció.
Flux alternatiu:	Cap

Cas d'ús	Consultar tots penyistes
Actors	Administrador
Objectiu	Veure tots els penyistes de la penya.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar a tots els penyistes que formen part de la penya. 2. Treu per pantalla una taula amb les dades resultants (id. penyista, nom, cognoms, data de naixement, NIF/NIE, e-mail, data d'alta, data de baixa i fons)
Flux alternatiu:	Si no hi ha cap penyista la taula surt buida.

Cas d'ús	Consultar penyistes grup primitiva
Actors	Administrador
Objectiu	Veure tots els penyistes que formen part del grup primitiva.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar als penyistes actius que formen part del grup primitiva. 2. Treu per pantalla una taula amb les dades resultants (id. penyista, nom, cognoms, data de naixement, NIF/NIE, e-mail, data d'alta i fons)
Flux alternatiu:	Si no hi ha cap penyista la taula surt buida.

Cas d'ús	Consultar penyistes grup euromilió
Actors	Administrador
Objectiu	Veure tots els penyistes que formen part del grup euromilió.

Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar als penyistes actius que formen part del grup euromilió. 2. Treu per pantalla una taula amb les dades resultants (id. penyista, nom, cognoms, data de naixement, NIF/NIE, e-mail, data d'alta i fons)
Flux alternatiu:	Si no hi ha cap penyista la taula surt buida.

Cas d'ús	Consultar totes apostes primitiva
Actors	Administrador
Objectiu	Veure tots les apostes primitives que hi ha a la BD.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar totes les apostes primitives. 2. Treu per pantalla una taula amb les dades resultants (id. aposta, tipus d'aposta, números combinació i si forma part del grup primitiva o no)
Flux alternatiu:	Si no hi ha cap aposta la taula surt buida.

Cas d'ús	Consultar totes apostes euromilió
Actors	Administrador
Objectiu	Veure tots les apostes d'euromilió que hi ha a la BD.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar totes les apostes d'euromilió. 2. Treu per pantalla una taula amb les dades resultants (id. aposta, tipus d'aposta, números combinació, estrelles i si forma part del grup euromilió o no)
Flux alternatiu:	Si no hi ha cap aposta la taula surt buida.

Cas d'ús	Consultar totes participacions
Actors	Administrador
Objectiu	Veure totes les participacions de la penya en els sorteigs de la primitiva i l'euromilió.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar totes les participacions. 2. Treu per pantalla una taula amb les dades resultants (id. participació, tipus d'aposta, id. bloc de penyista, id bloc d'aposta, data sorteig, validació, reintegraments en cas d'aposta primitiva, premis i id. sorteig)
Flux alternatiu:	Si no hi ha cap participació la taula surt buida.

Cas d'ús	Consultar participacions no validades
Actors	Administrador
Objectiu	Veure les participacions de la penya que encara no han estat validades per l'administració de loteries.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar les participacions que encara no s'han validat. 2. Treu per pantalla una taula amb les dades resultants (id. participació, tipus d'aposta, id. bloc de penyista, id bloc d'aposta, data sorteig, validació, reintegraments en cas d'aposta primitiva, premis i id. sorteig)
Flux alternatiu:	Si no hi ha cap participació la taula surt buida.

Cas d'ús	Consultar participacions sense sorteig
Actors	Administrador
Objectiu	Veure les participacions de la penya que encara no tenen associat un id. de sorteig.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar les participacion sense sorteig. 2. Treu per pantalla una taula amb les dades resultants (id. participació, tipus d'aposta, id. bloc de penyista, id bloc d'aposta, data sorteig, validació, reintegraments en cas d'aposta primitiva, premis i id. sorteig)
Flux alternatiu:	Si no hi ha cap participació la taula surt buida.

Cas d'ús	Consultar participacions sense premi
Actors	Administrador
Objectiu	Veure les participacions de la penya que encara no tenen associat el seu premi un cop sortejades.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar les participacion sense premi. 2. Treu per pantalla una taula amb les dades resultants (id. participació, tipus d'aposta, id. bloc de penyista, id bloc d'aposta, data sorteig, validació, reintegraments en cas d'aposta primitiva, premis i id. sorteig)
Flux alternatiu:	Si no hi ha cap participació la taula surt buida.

Cas d'ús	Consultar històric sorteigs primitiva
Actors	Administrador
Objectiu	Veure els resultats dels sortejos de primitiva introduïts per l'administrador, per tenir un històric a la penya i poder calcular la categoria de premi de cada participació.

Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar l'històric de sorteigs de la primitiva. 2. Treu per pantalla una taula amb les dades resultants (id. sorteig, data sorteig, validació, números combinació, complementari i reintegrament)
Flux alternatiu:	Si no hi ha cap participació la taula surt buida.

Cas d'ús	Consultar històric sorteigs euromilió
Actors	Administrador
Objectiu	Veure els resultats dels sortejos d'euromilió introduïts per l'administrador, per tenir un històric a la penya i poder calcular la categoria de premi de cada participació.
Precondició	Cap
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. El sistema accedeix a la BD per recuperar l'històric de sorteigs de l'euromilió. 2. Treu per pantalla una taula amb les dades resultants (id. sorteig, data sorteig, validació, números combinació i estrelles)
Flux alternatiu:	Si no hi ha cap participació la taula surt buida.

Cas d'ús	Crear/Seleccionar apostes grup
Actors	Administrador
Objectiu	Seleccionar una de les opcions disponibles per adreçar l'actor a la pàgina corresponent.
Precondició	L'actor ha d'estar validat.
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. L'actor selecciona una opció. 2. Pressiona el botó Crear/Sel 3. Anem a l'opció escollida.
Flux alternatiu:	Cap

Cas d'ús	Crear aposta primitiva
Actors	Administrador
Objectiu	Afegir una nova aposta primitiva a la BD de la penya per poder fer-la servir quan vulguem.
Precondició	Cap
Postcondició	S'actualiza la taula apuestas amb les noves dades.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix el nros. de l'aposta primitiva. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surt un missatge de confirmació
Flux alternatiu:	Si l'aposta està repetida es mostra un error.

Cas d'ús	Crear aposta euromilió
Actors	Administrador
Objectiu	Afegir una nova aposta euromilió a la BD de la penya per

	poder fer-la servir quan vulguem.
Precondició	Cap
Postcondició	S'actualiza la taula apuestas amb les noves dades.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix el nros. de l'aposta euromilió. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surt un missatge de confirmació.
Flux alternatiu:	Si l'aposta està repetida es mostra un error.

Cas d'ús	Seleccionar apostes grup primitiva
Actors	Administrador
Objectiu	Permet a l'actor afegir o treure apostes del bloc d'apostes que la penya fa servir per participar a la primitiva.
Precondició	Cap
Postcondició	S'actualitza la taula historapu.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix un id. d'aposta. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surten les dades de l'aposta. 5. Pressiona el botó (Añadir/Quitar) 6. Es repeteixen els passos anteriors per afegir o treure més apostes del bloc. 7. Un cop acabat es pressiona el botó Grabar cambios per emmagatzemar els canvis a la BD. 8. Es mostra un missatge de confirmació.
Flux alternatiu:	Si no hi ha canvis i hi ha errors es mostra missatge.

Cas d'ús	Seleccionar apostes grup euromilió
Actors	Administrador
Objectiu	Permet a l'actor afegir o treure apostes del bloc d'apostes que la penya fa servir per participar a l'euromilió.
Precondició	Cap
Postcondició	S'actualitza la taula historapu.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix un id. d'aposta. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surten les dades de l'aposta. 5. Pressiona el botó (Añadir/Quitar) 6. Es repeteixen els passos anteriors per afegir o treure més apostes del bloc. 7. Un cop acabat es pressiona el botó Grabar cambios per emmagatzemar els canvis a la BD. 8. Es mostra un missatge de confirmació.
Flux alternatiu:	Si no hi ha canvis i hi ha errors es mostra missatge.

Cas d'ús	Modificar fons penyista
Actors	Administrador
Objectiu	Permet a l'actor modificar el fons d'un penyista.
Precondició	Cap

Postcondició	S'actualitza la taula penyistas.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix un id. de penyista. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors mostra les dades del penyista. 5. S'emplena el camp per afegir o treure diners del fons. 6. Pressiona botó (Añadir/Quitar a Fondos) 7. Si hi ha errors es demana la seva correcció. 8. Si no hi ha errors mostra missatge de confirmació.
Flux alternatiu:	Cap

Cas d'ús	Modificar penyista grup
Actors	Administrador
Objectiu	Permet a l'actor afegir o treure un penyista dels grups que participen en els sorteigs del mes actual.
Precondició	Cap
Postcondició	S'actualitza la taula historpen.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix un id. de penyista. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surten les dades del penyista. 5. Es selecciona el grup o grups als que hi pertanyerà o deixarà de pertànyer el penyista. 6. Pressiona el botó Actualizar 7. Es mostra un missatge de confirmació.
Flux alternatiu:	Si no s'han fet canvis o hi ha hagut errors es mostra missatge.

Cas d'ús	Baixes
Actors	Administrador
Objectiu	Seleccionar una opció de les que hi ha disponibles a la secció de baixes.
Precondició	L'actor ha d'estar validat.
Postcondició	Cap
Escenari:	<ol style="list-style-type: none"> 1. L'actor selecciona la consulta desitjada. 2. Pressiona el botó Baja 3. Anem a l'opció escollida.
Flux alternatiu:	Cap

Cas d'ús	Baixa de penyista
Actors	Administrador
Objectiu	Permet a l'actor donar de baixa a un penyista de la penya.
Precondició	El penyista ha d'existir i no pot ser l'administrador.
Postcondició	S'actualitza la taula de penyistas
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix un id. de penyista. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surten les dades del penyista.

	5. Pressiona el botó Dar de baja 6. Es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Eliminar aposta
Actors	Administrador
Objectiu	Permet a l'actor treure una aposta de la BD.
Precondició	L'aposta no s'ha fet servir mai a cap participació.
Postcondició	S'actualitza la taula apuestas.
Escenari:	1. L'actor introdueix un id. d'aposta. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surten les dades de l'aposta. 5. Pressiona el botó Eliminar 6. Es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Participar sorteig primitiva
Actors	Administrador
Objectiu	Permet a l'actor generar una participació de la penya en un sorteig de la primitiva.
Precondició	Cap
Postcondició	S'actualitza la taula operaciones.
Escenari:	1. L'actor introdueix data de participació al sorteig. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Participar sorteig euromilió
Actors	Administrador
Objectiu	Permet a l'actor generar una participació de la penya en un sorteig de l'euromilió.
Precondició	Cap
Postcondició	S'actualitza la taula operaciones.
Escenari:	1. L'actor introdueix data de participació al sorteig. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Esborrar participació
Actors	Administrador
Objectiu	Permet a l'actor esborrar una participació de la BD.
Precondició	La participació encara no s'ha validat.
Postcondició	S'actualitza la taula operaciones.

Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix un id. de participació. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surten les dades de l'operació. 5. Pressiona el botó Borrar 6. Es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Validar/Invalidar participació
Actors	Administrador
Objectiu	Permet a l'actor activar o desactivar la validació d' una participació, i afegir els reintegraments en cas de sorteig de primitiva.
Precondició	No es pot invalidar una participació premiada.
Postcondició	S'actualitza la taula operaciones.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix un id. de participació. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surten les dades de l'operació. 5. Si està validada es pressiona el botó Invalidar 6. Si encara no està validada es pressiona el botó Validar, i en cas de ser primitiva s'afegeixen els reintegrament. 7. Si hi ha errors es demana la seva correcció. 8. Es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Afegir resultats sorteig primitiva
Actors	Administrador
Objectiu	Permet a l'administrador afegir a la BD els resultats d'un sorteig realitzat de la primitiva.
Precondició	No es pot afegir un sorteig que encara no s'ha realitzat.
Postcondició	S'actualitzen les taules sorteos i operaciones.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix la data de participació i els números del sorteig. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Afegir resultats sorteig euromilió
Actors	Administrador
Objectiu	Permet a l'administrador afegir a la BD els resultats d'un sorteig realitzat de l'euromilió.
Precondició	No es pot afegir un sorteig que encara no s'ha realitzat.
Postcondició	S'actualitzen les taules sorteos i operaciones.

Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix la data de participació i els números del sorteig. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Afegir premis participació
Actors	Administrador
Objectiu	Permet a l'actor afegir els premis obtinguts a una participació.
Precondició	La participació ha d'estar validada i amb un sorteig assignat.
Postcondició	S'actualitza les taules penyistas i operaciones.
Escenari:	<ol style="list-style-type: none"> 1. L'actor introdueix un id. de participació. 2. Pressiona Aceptar 3. Si hi ha errors es demana la seva correcció. 4. Si no hi ha errors surten les dades de l'operació. 5. Introdueix la quantia premiada. 6. Pressiona el botó Añadir Premio 7. Si hi ha errors es demana la seva correcció. 8. Es mostra un missatge de confirmació.
Flux alternatiu:	Si hi ha hagut errors es mostra missatge.

Cas d'ús	Informes
Actors	Administrador
Objectiu	Permet a l'actor obtenir l'informe seleccionat en el format que ha escollit.
Precondició	L'actor ha d'estar validat.
Postcondició	Es genera l'informe en el format desitjat (PDF, excel o word)
Escenari:	<ol style="list-style-type: none"> 1. L'actor selecciona un informe. 2. Escull el format de sortida. 3. Pressiona el botó Aceptar
Flux alternatiu:	Cap

C. Arquitectura multicapa amb n-nivells (aplicació JEE)

JEE estén l'arquitectura típica de tres capes o nivells per definir una arquitectura de n-capes.

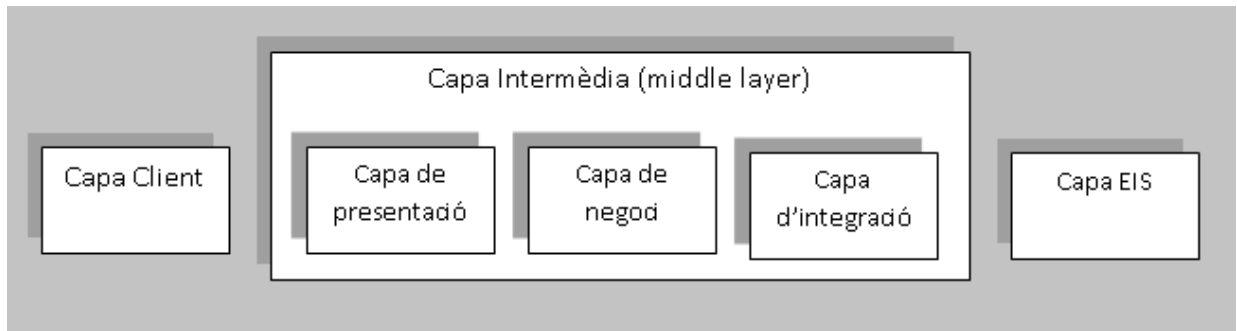


Figura C.1 Arquitectura multicapa amb n-nivells

Capa client

Conté els components que s'executen en les màquines client. Per exemple, pàgines HTML que s'executen en els navegadors dels usuaris.

Una aplicació JEE pot tenir quatre tipus fonamentals de clients:

- Les miniaplicacions (applets): programes fets en Java que s'executen en la màquina virtual que tenen els navegadors i que es descarreguen dels servidors per ser executats en els clients.
- Els clients web (thin client): és el tipus de client per excel·lència en les aplicacions JEE.
- Els clients MIDlet (JME): petites aplicacions Java que s'executen en dispositius mòbils.
- Els clients d'escriptori (rich client): s'executen en una màquina virtual instal·lada a l'ordinador del client i accedeixen directament als components EJB de la capa de negoci del servidor.

Capa intermèdia (middle layer)

Els components d'aquesta capa s'executen en els diferents contenidors que formen el servidor d'aplicacions. Podem dividir la capa intermèdia en tres subcapes formades per components diferents que s'executen en contenidors diferents i que tenen funcionalitats diferents:

Capa de presentació

- Permetre als clients que cridin a la lògica de negoci de l'aplicació recollint les peticions d'aquests, invocant al mètode de negoci corresponent i presentant els resultats als clients.
- Generar contingut dinàmic.
- Gestionar el flux de presentació, és a dir, donada una entrada i un resultat en l'execució de la lògica de negoci, decidir quina serà la següent pantalla que cal mostrar a l'usuari.
- Fer certes validacions en les dades que rep com a entrada.
- Mantenir l'estat entre diferents peticions sobre el protocol http (que no té estat)
- Donar suport a diversos tipus de clients. Per exemple, la capa web d'una aplicació JEE podria servir contingut a clients lleugers amb HTML i a telèfons mòbils amb WML.
- Conté els components (servlets, jsp) de presentació que s'executen al contenidor web del servidor d'aplicacions.
- En aquest àmbit d'actuació s'apliquen tecnologies com Tomcat, Java Server Faces, Struts, Spring MVC, ...

Capa de negoci

- Conté el codi i les regles de negoci de la nostra aplicació.
- JEE defineix els components de negoci com EJB's de sessió que s'executen al contenidor de EJB's del servidor d'aplicacions.
- En aquest àmbit d'actuació s'apliquen tecnologies com Jboss, Glassfish, Spring, ...
- Poden haver-hi classes POJO.

Capa d'integració

- Capa encarregada d'accedir i tractar les dades que poden estar emmagatzemats en fonts heterogènies (bases de dades relacionals, sistemes heretats, bases de dades orientades a objecte, ERP, etc). Posteriorment passa les dades a la capa de negoci perquè aquesta faci les transformacions pertinents en cada cas.
- Un dels components de la capa d'integració que defineix JEE són els EJB d'entitat, que també s'executen al contenidor de EJB's del servidor d'aplicacions.
- En aquest àmbit d'actuació s'apliquen tecnologies com JPA, Hibernate, iBatis, JDBC, ...
- Poden haver-hi classes POJO.

Capa EIS (Enterprise Information System)

En aquesta capa hi trobem la resta de components del sistema d'informació als quals s'accedirà des de la capa d'integració.

Aquests sistemes són totalment heterogenis i normalment separats físicament de les altres capes. Poden anar des de servidors de bases de dades relacionals o orientades a objectes, ERP, sistemes SAP, sistemes de fitxers, etc.

Altres arquitectures per a aplicacions JEE

Arquitectura amb components no distribuïts i EJS's de sessió:

- Capa de presentació amb servlets i JSP seguint un patró MVC.
- Accés local de la capa de presentació a la capa de negoci.
- Façana d'accés a la capa de negoci amb EJB's de sessió sense estat i local.
- Capa d'integració normalment desenvolupada sense utilitzar EJB's d'entitat, sinó amb algun framework de mapeig objecte/relacional.

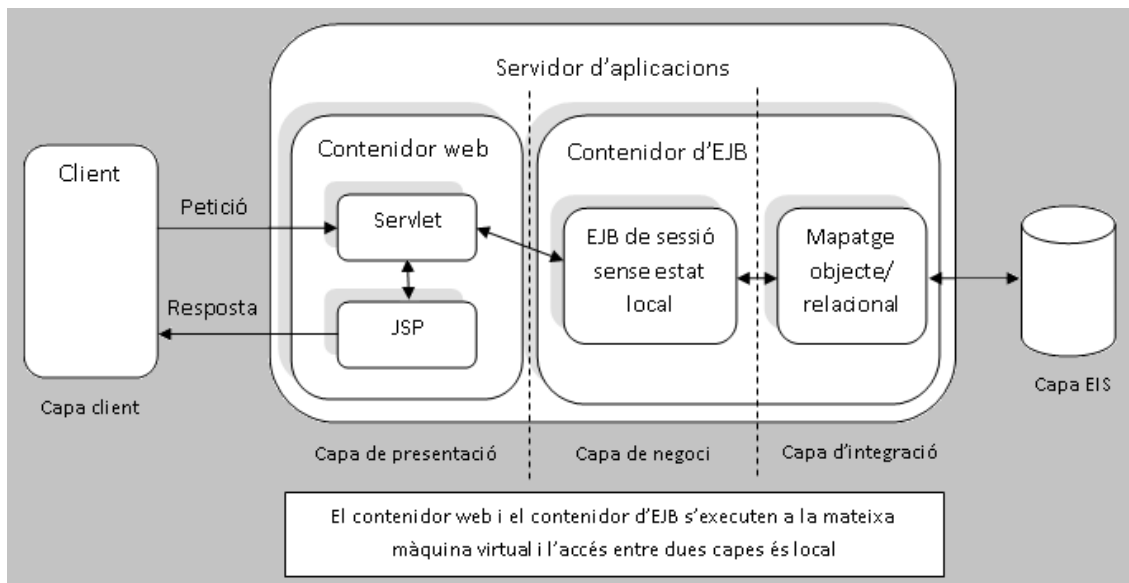


Figura C.2 Arquitectura JEE no distribuïda amb EJB's de sessió

Arquitectura amb components distribuïts:

- Capa de presentació amb servlets i JSP seguint un patró MVC.
- Accés remot de la capa de presentació a la capa de negoci mitjançant RMI.
- Façana d'accés a la capa de negoci amb EJB's de sessió sense estat i remot.
- Capa d'integració utilitzant EJB's d'entitat amb CMP i accedits localment des dels EJB de sessió que fan de façana.

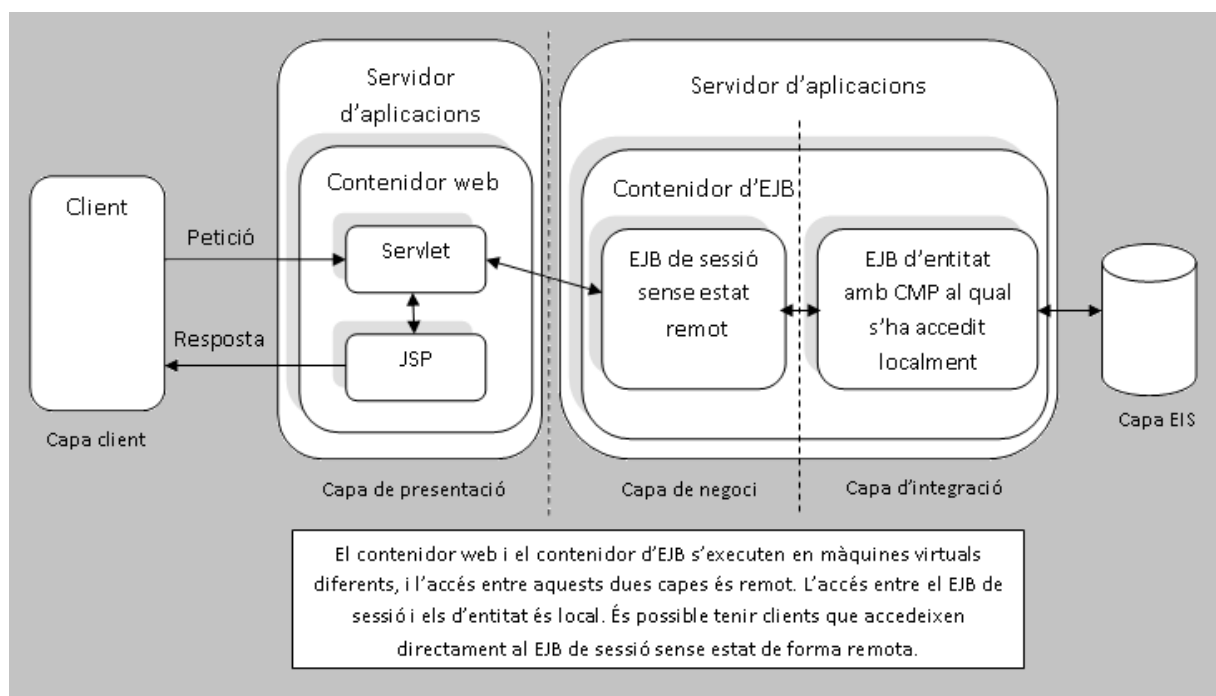


Figura C.3 Arquitectura JEE distribuïda

D. Llibreries utilitzades a Eclipse

Les llibreries que s'han incorporat al buildpath del projecte d'Eclipse (directori `\WebContent\WEB-INF\lib`) son:

- Controlador JDBC proporcionat per MySQL:
mysql-connector-java-5.1.12-bin.jar
- Llibreries de JasperReports:
bsh-2.0b4.jar
commons-beanutils-1.8.0.jar
commons-collections-2.1.1.jar
commons-digester-1.7.jar
commons-javafLOW-20060411.jar
commons-logging-1.0.4.jar
groovy-all-1.5.5.jar
iText-2.1.7.jar
jasperreports-3.7.1.jar
poi-3.5-FINAL-20090928.jar
- Llibreries pel framework de persistència EclipseLink:
eclipselink.jar
javax.persistence_1.0.0.jar
- Llibreries per la biblioteca d'etiquetes estàndard per JSP (JSTL):
jstl.jar
standard.jar
- Llibreria pel framework de paginació pager-taglib-2.0:
pager-taglib.jar
- Llibreries per la generació de gràfiques en Java amb JfreeChart:
jcommon-1.0.17.jar
jfreechart-1.0.14.jar
- Llibreria del framework de presentació SiteMesh:
sitemesh-2.4.1.jar

Les llibreries de JavaScript incorporades al directori `\WebContent\js` d'Eclipse son:

- Llibreries JQuery per la validació de camps dels formularis:

jquery-1.8.0.js

jquery.validate.js

jquery.numeric.js

- Llibreries per la generació del calendari:

calendar.js

calendar-setup.js

calendar-es.js

E. Passos per fer l'aplicació web segura

- Instal·lació de ActivePerl, versió gratuïta per a sistemes Windows, i del programa OpenSSL.
- Creació d'un directori de treball (*c:\certificats*)
- Creació d'una variable d'entorn, el valor és la localització del fitxer 'openssl.cnf' (*\OpenSSL\bin*)

- Modificar una línia del fitxer 'openssl.cnf':

And for everything including object signing:

nsCertType = client, email, objsign, server

- Per crear l'Autoritat Certificadora (CA), des del directori de treball:

c:\certificats>perl c:\openssl\bin\ca.pl -newca

(la comanda anterior genera un subdirectori anomenat demoCA)

- Editem *demoCA\cacert.pem*, eliminem les línies inicials fins a la línia que conté BEGIN CERTIFICATE i renombram a *demoCA\cacertTomcat.pem*

(Ara generem el certificat signat pel servidor)

- Generar el parell de claus per Tomcat (el CN de Tomcat ha de ser el nom DNS del servidor en què s'executi):

c:\certificats>keytool -genkey -alias tomcat -keyalg RSA -keystore

myKeyStoreOpenSSL

- Generar una sol·licitud de certificat partint del parell de claus generat en el pas previ:
`c:\certificats>keytool -certreq -alias tomcat -file newreq.pem -keystore myKeyStoreOpenSSL`
- Signar la sol·licitud de certificat amb la CA (es genera el fitxer 'newcert.pem'):
`c:\certificats>perl c:\openssl\bin\ca.pl -sign`
- Editem 'newcert.pem', eliminant les línies inicials fins a la línia que conté BEGIN CERTIFICATE i el guardem com 'tomcat.pem'
- Instal·lem el certificat autosignat de la CA al magatzem de certificats de CA que farà servir Tomcat:
`c:\certificats>keytool -import -trustcacerts -alias projecteCA -file demoCA\cacertTomcat.pem -keystore myKeyStoreOpenSSL`
- I per acabar instal·lem el certificat signat per la CA en el keystore:
`c:\certificats>keytool -import -trustcacerts -alias tomcat -file tomcat.pem -keystore myKeyStoreOpenSSL`
- Per activar l'ús de SSL a Tomcat es faran canvis al fitxer 'server.xml' (*TOMCAT_HOME/conf*):
 (deixar sense comentaris l'element Connector)

```
<Connector port = "8443" protocol = "org.apache.coyote.http11.Http11Protocol"
  SSLEnabled = "true" maxThreads = "150" scheme = "https" secure = "true"
  maxHttpHeaderSize = "8192" minSpareThreads = "25"
  maxSpareThreads = "75" acceptCount = "100" enableLookups = "false"
  disableUploadTimeout = "true" clientAuth = "false" sslProtocol = "TLS"
  truststoreFile = "C: \ Certificats \ myKeyStoreOpenSSL" truststorePass = "euoprim"
  keystoreFile = "C: \ Certificats \ myKeyStoreOpenSSL" keystorePass = "euoprim" />
```
- Perquè el navegador reconegui la CA i, per tant, pugui verificar la validesa del certificat de Tomcat, cal instal·lar el certificat de la CA en el propi navegador.

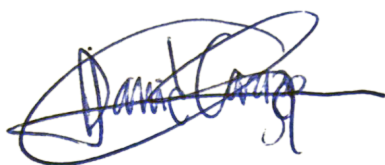
Bibliografia

- Documentació subministrada al curs Analista-Programador en Java, realitzat al centre C.I.F.O. la Violeta (Març 2010)

Referencies web

- [1] Webs de hosting consultades <http://www.hostigal.com>, <http://www.anw.es>, <http://www.arvixe.com>,...
- [2] Webs que ofereixen certificats SSL <http://www.thawte.com>, <http://www.globalsign.com>, <http://ssl.comodo.com>, <http://www.verisign.com>,...
- [3] Object Management Group (OMG) <http://www.omg.org>
- Documentació de productes Oracle
<http://www.oracle.com/technetwork/indexes/documentation/index.html>
- Entorn de proves PayPal
<https://developer.paypal.com/webapps/developer/applications/accounts>
- Etiquetes HTML <http://www.htmlquick.com/es/reference/tags.html>
- JasperReports 3.7.1 API
<http://www.jarvana.com/jarvana/view/net/sf/jasperreports/jasperreports/3.7.1/jasperreports-3.7.1-javadoc.jar!/index.html>
- Java EE 6 API Specification <http://docs.oracle.com/javaee/6/api/>
- Java SE 7 API Specification <http://docs.oracle.com/javase/7/docs/api/>
- jQuery API Documentation <http://api.jquery.com/>
- JSTP 1.1 API Specification <http://docs.oracle.com/javaee/5/jstl/1.1/docs/tlddocs/>

- Manuals de CSS, HTML, JavaScript, Java, BBDD,...
<http://www.jorgesanchez.net/index.html>
- MySQL 5.0 Reference Manual <http://dev.mysql.com/doc/#manual>
- OpenSSL project <http://www.openssl.org/>
- Referècies de CSS 2.1 http://librosweb.es/referencia_css/
- The Java EE 6 Tutorial <http://docs.oracle.com/javaee/6/tutorial/doc/>
- W3Schools Online Web Tutorials [en línea] <http://www.w3schools.com/>
- Wikipedia, la enciclopedia libre <http://www.wikipedia.org>
- World Wide Web Consortium (W3C) <http://www.w3.org/>



Signat: David Carazo Herrera
Bellaterra, 18 de Juny de 2013